Hamid Sarbazi-Azad
Behrooz Parhami
Seyed-Ghassem Miremadi
Shaahin Hessabi (Eds.)

Communications in Computer and Information Science 6

Advances in Computer Science and Engineering

13th International CSI Computer Conference, CSICC 2008
Kish Island, Iran, March 2008
Revised Selected Papers

Springer

Communications
in Computer and Information Science 6

Hamid Sarbazi-Azad Behrooz Parhami
Seyed-Ghassem Miremadi Shaahin Hessabi (Eds.)

Advances in Computer Science and Engineering

13th International CSI Computer Conference, CSICC 2008
Kish Island, Iran, March 9-11, 2008
Revised Selected Papers

Springer

Volume Editors

Hamid Sarbazi-Azad
Seyed-Ghassem Miremadi
Shaahin Hessabi
Sharif University of Technology
Department of Computer Engineering
Azadi Street, Tehran, Iran
E-mail: {azad, miremadi, hessabi}@sharif.edu
and
Hamid Sarbazi-Azad
Institute for Studies in Theoretical Physics and Mathematics
School of Computer Science
Niavaran Square, Tehran, Iran
E-mail: azad@ipm.ir

Behrooz Parhami
University of California, Santa Barbara
Department of Electrical and Computer Engineering
Santa Barbara, CA 93106-9560, USA
E-mail: parhami@ece.ucsb.edu

Library of Congress Control Number: 2008940423

CR Subject Classification (1998): H.2.8, I.2, H.5, I.5, C.5

ISSN 1865-0929
ISBN-10 3-540-89984-7 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-89984-6 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12587735 06/3180 5 4 3 2 1 0

Preface

It is our pleasure to welcome you to the proceedings of the 13th International Computer Society of Iran Computer Conference (CSICC-2008). The conference has been held annually since 1995, except for 1998, when it transitioned from a year-end to first-quarter schedule. It has been moving in the direction of greater selectivity (see Fig.1) and broader international participation. Holding it in Kish Island this year represents an effort to further facilitate and encourage international contributions. We feel privileged to participate in further advancing this strong technical tradition.

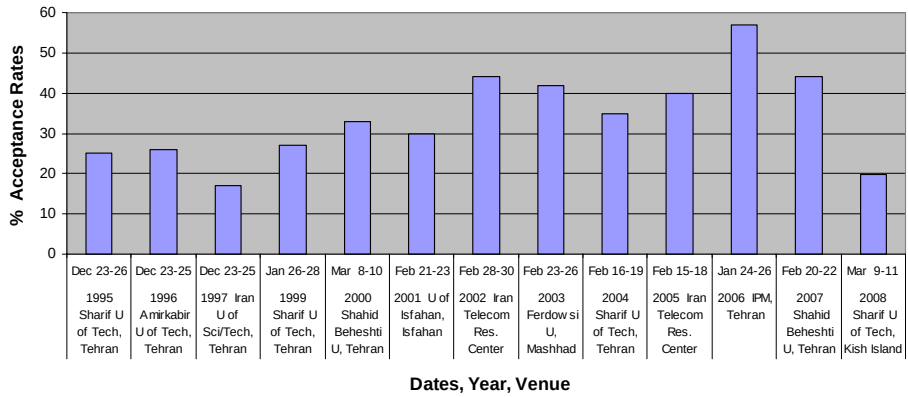


Fig. 1. The CSI Computer Conferences: Dates, Venues, and Acceptance Rates for Regular Papers

This year 426 papers were submitted to CSICC 2008, of which 84 were accepted as regular papers and 68 as posters (short papers). The task of handling such a large number of papers was both rewarding and challenging. Compiling this year's technical program would not have been possible without the enthusiastic participation of 1,140 authors, 374 reviewers, 32 Program Committee members, 27 Organizing Committee members, 15 program/organizing committees assistants, and 19 sponsoring organizations. The conference proceedings are published by Springer in the series "Communications in Computer and Information Science" (CCIS). To all these contributors, we owe a debt of gratitude. We are especially indebted to the primary sponsor of the event, Sharif University of Technology.

A technical conference contributes to the exchange of new ideas via formal presentations and proceedings papers. However, it is the informal technical exchanges and networking among participants that offer the greatest potential benefits. We hope that all CSICC 2008 participants could take advantage of this gathering opportunity, in the

serene environment of Kish Island, to acquaint themselves with new ideas and to plant the seeds of collaborative projects with colleagues from other institutions or countries.

Hamid Sarbazi-Azad
Behrooz Parhami,
Seyed-Ghassem Miremadi
Shaahin Hessabi

Organization

General Co-chairs

Shaahin Hessabi	Sharif University of Technology, Tehran, Iran
Seyed-Ghassem Miremadi	Sharif University of Technology, Tehran, Iran

Program Co-chairs

Behrooz Parhami	University of California, Santa Barbara, CA, USA
Hamid Sarbazi-Azad	Sharif University of Technology and IPM, Tehran, Iran

Track Chairs

Algorithms and Theory of Computing	M. Ghodsi (Iran) M. Mahdian (USA)
Artificial Intelligence and Learning	Badie (Iran) A. Khotanzad (USA)
Computer Architecture and Arithmetic	N. Bagherzadeh (USA) A. Jahangir (Iran)
Computer Networks and Data Communications	H. Perdram (Iran) A. Jamalipour (USA)
Computer Security and Cryptography	R. Jalili (Iran) A. Ghorbani (Canada)
Computer Vision, Image Processing and Graphics	S. Kasaei (Iran) B. Boashash (Australia)
Dependability, Fault Tolerance and Testability	G. Miremadi (Iran) H. Madeira (Potugal)
Information Technology and Enterprise Architecture	R. Ayani (Sweden) J. Habibi (Iran)
Internet, Grid and Cluster Computing	M. Yaghmaei (Iran) A. Shahrabi (UK)
Parallel Processing and Distributed Systems	M. Meybodi (Iran) M. Ould-khaoua (Oman)
Performance Modeling and Evaluation	I. Awan (UK) A. Khonsari (Iran)
Soft-Computing	S. Bagheri (Iran) H. Berenji (USA)

Software Engineering and Formal Methods	F. Arbab (The Netherlands) A. Movaghar (Iran)
Speech and Signal Processing	H. Sameti (Iran) S. Boussakta (UK)
VLSI	S. Hessabi (Iran) M. Pedram (USA)
Advanced Topics in Computer Science and Engineering	B. Parhami (USA) H. Sarbazi-Azad (Iran)

Table of Contents

Full Papers

Learning/Soft Computing

A Novel Piecewise Linear Clustering Technique Based on Hyper Plane Adjustment 1
Mohammad Taheri, Elham Chitsaz, Seraj D. Katebi, and Mansoor Z. Jahromi

Ant Colony Optimization with a Genetic Restart Approach toward Global Optimization 9
G. Hossein Hajimirsadeghi, Mahdy Nabaee, and Babak N. Araabi

Automatic Extraction of IS-A Relations in Taxonomy Learning 17
Mahmood Neshati, Hassan Abolhassani, and Hassan Fatemi

A Bayesian Network Based Approach for Data Classification Using Structural Learning 25
A.R. Khanteymoori, M.M. Homayounpour, and M.B. Menhaj

A Clustering Method Based on Soft Learning of Model (Prototype) and Dissimilarity Metrics 33
Arash Arami and Babak Nadjar Araabi

Algorithm and Theory (1)

An Approximation Algorithm for the k-Level Uncapacitated Facility Location Problem with Penalties 41
Mohsen Asadi, Ali Niknafs, and Mohammad Ghodsi

Optimizing Fixpoint Evaluation of Logic Programs with Uncertainty 50
Nematollaah Shiri and Zhi Hong Zheng

Approximating Component Selection with General Costs 61
Mostafa Nouri and Jafar Habibi

Online Suffix Tree Construction for Streaming Sequences 69
Giyasettin Ozcan and Adil Alpkocak

Fuzzy Voronoi Diagram 82
Mohammadreza Jooyandeh and Ali Mohades Khorasani

SoC and NoC

A Novel Partitioned Encoding Scheme for Reducing Total Power Consumption of Parallel Bus 90
Mehdi Kamal, Somayyeh Koohi, and Shaahin Hessabi

Efficient Parallel Buffer Structure and Its Management Scheme for a Robust Network-on-Chip (NoC) Architecture 98
Jun Ho Bahn and Nader Bagherzadeh

Integration of System-Level IP Cores in Object-Oriented Design Methodologies 106
Shoaleh Hashemi Namin and Shaahin Hessabi

Polymorphism-Aware Common Bus in an Object-Oriented ASIP 115
Nima Karimpour Darav and Shaahin Hessabi

Efficient VLSI Layout of WK-Recursive and WK-Pyramid Interconnection Networks 123
Saeedeh Bakhshi and Hamid Sarbazi-Azad

Wireless/Sensor Networks

Energy Adaptive Cluster-Head Selection for Wireless Sensor Networks Using Center of Energy Mass 130
Ehsan Akhtarkavan and Mohammad Taghi Manzuri Shalmani

SHRP: A New Routing Protocol to Wireless Sensor Networks 138
Cláudia J. Barenco Abbas, Nelson Cárdenas, Giácomo Lobalsamo, and Néstor Davila

Improvement of MAC Performance for Wireless Sensor Networks 147
M.H. Fotouhi Ghazvini, M. Vahabi, M.F.A. Rasid, and R.S.A. Raja Abdullah

Route Optimization Security in Mobile IPv6 Wireless Networks: A Test-Bed Experience 153
Abbas Mehdizadeh, S. Khatun, Borhanuddin M. Ali, R.S.A. Raja Abdullah, and Gopakumar Kurup

Adaptive End-to-End QoS for Multimedia over Heterogeneous Wireless Networks 160
Ouldooz Baghban Karimi, Mahmood Fathy, and Saleh Yousefi

A Blocking Reduction Scheme for Multiple Slot Cell Scheduling in Multicast Switching Systems 168
Fong-Cheng Lee, Wen-Fong Wang, and Jung-Bin Shih

Video Processing and Related Topics

Object-Based Video Coding for Distance Learning Using Stereo Cameras 176
Amir Hossein Khalili, Mojtaba Bagheri, and Shohreh Kasaei

Seabed Image Texture Segmentation and Classification Based on Nonsubsampled Contourlet Transform 186
Reza Javidan, Mohammad A. Masnadi-Shirazi, and Zohreh Azimifar

Unequal Error Protection for the Scalable Extension of H.264/AVC Using Genetic Algorithm 194
Amir Naghdinezhad, Mahmoud Reza Hashemi, and Omid Fatemi

An Adaptive Method for Moving Object Blending in Dynamic Mosaicing 203
Mojtaba Bagheri, Tayebeh Lotfi, and Shohreh Kasaei

Inferring a Bayesian Network for Content-Based Image Classification ... 211
Shahriar Shariat, Hamid R. Rabiee, and Mohammad Khansari

High Performance Mathematical Quarter-Pixel Motion Estimation with Novel Rate Distortion Metric for H.264/AVC 219
Somayeh Sardashti, Hamid Reza Ghasemi, Mehdi Semsarzadeh, and Mahmoud Reza Hashemi

Processor Architecture

A Versatile Reconfigurable Bit-Serial Multiplier Architecture in Finite Fields GF(2m) 227
Morteza Nikooghadam, Ehsan Malekian, and Ali Zakerolhosseini

A Nonspeculative Maximally Redundant Signed Digit Adder 235
Ghassem Jaberipur and Saeid Gorgin

System-Level Assertion-Based Performance Verification for Embedded Systems 243
Hassan Hatefi-Ardakani, Amir Masoud Gharehbaghi, and Shaahin Hessabi

The Effect of Core Number and Core Diversity on Power and Performance in Multicore Processors 251
A. Zolfaghari Jooya and M. Soryani

Reducing the Computational Complexity of an RLS-Based Adaptive Controller in ANVC Applications 259
Allahyar Montazeri and Javad Poshtan

An Efficient and Extendable Modeling Approach for VLIW DSP Processors 267
Naser Sedaghati-Mokhtari, Mahdi Nazm-Bojnordi, Abbas Hormati, and Sied Mehdi Fakhraie

Algorithm and Theory (2)

An Exact Algorithm for the Multiple-Choice Multidimensional Knapsack Based on the Core 275
Mohammad Reza Razzazi and Taha Ghasemi

Kinetic Polar Diagram 283
Mojtaba Nouri Bygi, Fatemeh Chitforoush, Maryam Yazdandoost, and Mohammad Ghodsi

A Graph Transformation-Based Approach to Formal Modeling and Verification of Workflows 291
Vahid Rafe and Adel T. Rahmani

Efficient Parallel Routing Algorithms for Cartesian and Composition Networks 299
Marzieh Bakhshi, Saeedeh Bakhshi, and Hamid Sarbazi-Azad

AI/Robotics/Control

A Naïve Bayes Classifier with Distance Weighting for Hand-Gesture Recognition 308
Pujan Ziaie, Thomas Müller, Mary Ellen Foster, and Alois Knoll

SBUQA Question Answering System 316
Mahsa A. Yarmohammadi, Mehrnoush Shamsfard, Mahshid A. Yarmohammadi, and Masoud Rouhizadeh

A New Feedback ANC System Approach 324
Pooya Davari and Hamid Hassanpour

Benefiting White Noise in Developing Feedforward Active Noise Control Systems 332
Pooya Davari and Hamid Hassanpour

Medical Image Processing

Comparison of Linear and Nonlinear Models for Estimating Brain Deformation Using Finite Element Method 340
Hajar Hamidian, Hamid Soltanian-Zadeh, Alireza Akhondi-Asl, and Reza Faraji-Dana

Context-Dependent Segmentation of Retinal Blood Vessels Using Hidden Markov Models 348
Amir Pourmorteza, Seyed Hamid Reza Tofighi, Alireza Roodaki, Ashkan Yazdani, and Hamid Soltanian-Zadeh

Retinal Vessel Extraction Using Gabor Filters and Support Vector Machines 356
Alireza Osareh and Bita Shadgar

A New Segmentation Method for Iris Recognition Using the Complex Inversion Map and Best-Fitting Curve 364
Sepehr Attarchi, Karim Faez, and Mir Hashem Mousavi

P2P, Cluster, and Grid Systems

A New Algorithm for Combating Free-Riders in Structured P2P Networks 372
Mohammad R. Raeesi N., Jafar Habibi, Pariya Raoufi, and Habib Rostami

Clustering Search Engine Log for Query Recommendation 380
Mehdi Hosseini and Hassan Abolhassani

Reliability Evaluation in Grid Environment 388
Saeed Parsa and Fereshteh Azadi Parand

Formulating Priority Coefficients for Information Quality Criteria on the Blog 396
Mohammad Javad Kargar, Abdul Rahman Ramli, H. Ibrahim, and F. Azimzadeh

A Distributed Proxy System for High Speed Clients 404
Martin Krohn, Helena Unger, and Djamshid Tavangarian

Mobile Ad Hoc Networks

New Routing Strategies for RSP Problems with Concave Cost 412
Marjan Momtazpour and Pejman Khadivi

Modeling Routing Protocols in Adhoc Networks 419
Fatemeh Ghassemi and Ali Movaghar

CEBAC: A Decentralized Cooperation Enforcement Based Access Control Framework in MANETs 427
Fatemeh Saremi, Hoda Mashayekhi, Ali Movaghar, and Rasool Jalili

A Secure Cross-Layer Design of Clustering-Based Routing Protocol for MANET 435
Arash Dana and Marzieh Hajhosseini

Web

An Approach for Semantic Web Query Approximation Based on Domain Knowledge and User Preferences 443
Zeinab Iranmanesh, Razieh Piri, and Hassan Abolhassani

Semantic Web Services for Handling Data Heterogeneity in an E-Business Framework 453
Seyyed Ali Rokni Dezfouli, Jafar Habibi, and Soheil Hassas Yeganeh

Challenges in Using Peer-to-Peer Structures in Order to Design a Large-Scale Web Search Engine 461
Hamid Mousavi and Ali Movaghar

Signal Processing/Speech Processing

Sparse Sinusoidal Signal Representation for Speech and Music Signals 469
Pejman Mowlaee, Amirhossein Froghani, and Abolghasem Sayadiyan

Variant Combination of Multiple Classifiers Methods for Classifying the EEG Signals in Brain-Computer Interface 477
Zahra Shoaie Shirehjini, Saeed Bagheri Shouraki, and Maryam Esmailee

Nevisa, a Persian Continuous Speech Recognition System 485
Hossein Sameti, Hadi Veisi, Mohammad Bahrani, Bagher Babaali, and Khosro Hosseinzadeh

Effects of Feature Domain Normalizations on Text Independent Speaker Verification Using Sorted Adapted Gaussian Mixture Models 493
Rahim Saeidi, Hamid Reza Sadegh Mohammadi, Todor Ganchev, and Robert D. Rodman

A Centrally Managed Dynamic Spectrum Management Algorithm for Digital Subscriber Line Systems 501
Adnan Rashdi, Noor Muhammad Sheikh, and Asrar ul Haq Sheikh

Misc.

Methods for Analyzing Information Contained in an Enterprise Email Database 509
Mohsen Sadeghi, Khaled Hadj-Hamou, and Mickaël Gardoni

Early Bug Detection in Deployed Software Using Support Vector Machine 518
Saeed Parsa, Somaye Arabi Nare, and Mojtaba Vahidi-Asl

A New Architecture for Heterogeneous Context Based Routing 526
Enrico Dressler, Raphael Zender, Ulrike Lucke, and Djamshid Tavangarian

Performance Modeling of a Distributed Web Crawler Using Stochastic Activity Networks 535
Mitra Nasri, Saeed Shariati, and Mohammad Abdollahi Azgomi

Performance Comparison of Simple Regular Meshes and Their k-ary n-cube Variants in Optical Networks 543
Ahmad Kianrad, Aresh Dadlani, Ali Rajabi, Mohammadreza Aghajani, Ahmad Khonsari, and Seyed Hasan Seyed Razi

Security

A Robust and Efficient SIP Authentication Scheme 551
Alireza Mohammadi-nodooshan, Yousef Darmani, Rasool Jalili, and Mehrdad Nourani

A Temporal Semantic-Based Access Control Model 559
Ali Noorollahi Ravari, Morteza Amini, and Rasool Jalili

A Review on Concepts, Algorithms and Recognition-Based Applications of Artificial Immune System 569
Shahram Golzari, Shyamala Doraisamy, Md Nasir B. Sulaiman, and Nur Izura Udzir

Incremental Hybrid Intrusion Detection Using Ensemble of Weak Classifiers 577
Amin Rasoulifard, Abbas Ghaemi Bafghi, and Mohsen Kahani

A Cluster-Based Key Establishment Protocol for Wireless Mobile Ad Hoc Networks 585
Mohammad Sheikh Zefreh, Ali Fanian, Sayyed Mahdi Sajadieh, Pejman Khadivi, and Mehdi Berenjkoub

Time Series Analysis for ARP Anomaly Detection: A Combinatorial Network-Based Approach Using Multivariate and Mean-Variance Algorithms 593
Yasser Yasami, Saadat Pourmozaffari, and Siavash Khorsandi

Image Processing Applications

Polyhedral GPU Accelerated Shape from Silhouette 601
Alireza Haghshenas, Mahmoud Fathy, and Maryam Mokhtari

Discrimination of Bony Structures in Cephalograms for Automatic Landmark Detection 609
Rahele Kafieh, Saeed Sadri, Alireza Mehri, and Hamid Raji

Grid Based Registration of Diffusion Tensor Images Using Least Square Support Vector Machines 621
Esmaeil Davoodi-Bojd and Hamid Soltanian-Zadeh

Detection of Outer Layer of the Vessel Wall and Characterization of Calcified Plaques in IVUS Images 629
Alireza Roodaki, Zahra Najafi, Armin Soltanzadi, Arash Taki, Seyed Kamaledin Setarehdan, and Nasir Navab

Object Modeling for Multicamera Correspondence Using Fuzzy Region Color Adjacency Graphs 637
Amir Hossein Khalili and Shohreh Kasaei

Secure Digital Image Watermarking Based on SVD-DCT 645
Azadeh Mansouri, Ahmad Mahmoudi Aznaveh, and Farah Torkamani Azar

VLSI

A Novel Delay Fault Testing Methodology for Resistive Faults in Deep Sub-micron Technologies 653
Reza Javaheri and Reza Sedaghat

On the Importance of the Number of Fanouts to Prevent the Glitches in DPA-Resistant Devices 661
Amir Moradi, Mahmoud Salmasizadeh, and Mohammad Taghi Manzuri Shalmani

Performance Enhancement of Asynchronous Circuits 671
Somaye Raoufifard, Behnam Ghavami, Mehrdad Najibi, and Hossein Pedram

A Low Power SRAM Based on Five Transistors Cell 679
Arash Azizi Mazreah and Mohammad Taghi Manzuri Shalmani

Evaluating the Metro-on-Chip Methodology to Improve the Congestion and Routability 689
Ali Jahanian, Morteza Saheb Zamani, Mostafa Rezvani, and Mehrdad Najibi

Sequential Equivalence Checking Using a Hybrid Boolean-Word Level Decision Diagram 697
Bijan Alizadeh and Masahiro Fujita

A Circuit Model for Fault Tolerance in the Reliable Assembly of Nano-Systems 705
Masoud Hashempour, Zahra Mashreghian Arani, and Fabrizio Lombardi

Short Papers

An Operator for Removal of Subsumed Clauses 714
Mohammad Ghasemzadeh and Christoph Meinel

Low Power and Storage Efficient Parallel Lookup Engine Architecture for IP Packets 718
Alireza Mahini, Reza Berangi, Hossein Mohtashami, and Hamidreza Mahini

Assignment of OVSF Codes in Wideband CDMA 723
Mehdi Askari, Reza Saadat, and Mansour Nakhkash

Toward Descriptive Performance Model for OpenMosix Cluster 728
Bestoun S. Ahmed, Khairulmizam Samsudin, Abdul Rahman Ramli, and ShahNor Basri

Analysis of the Growth Process of Neural Cells in Culture Environment Using Image Processing Techniques 732
Atefeh S. Mirsafian, Shirin N. Isfahani, Shohreh Kasaei, and Hamid Mobasheri

Bandwidth-Delay Constrained Least Cost Multicast Routing for Multimedia Communication 737
Mehrdad Mahdavi, Rana Forsati, and Ali Movaghar

Cellular Probabilistic Evolutionary Algorithms for Real-Coded Function Optimization 741
M.R. Akbarzadeh T. and M. Tayarani N.

A New Approach for Scoring Relevant Documents by Applying a Farsi Stemming Method in Persian Web Search Engines 745
Hamed Shahbazi, Alireza Mokhtaripour, Mohammad Dalvi, and Behrouz Tork Ladani

FRGA Matching Algorithm in High-Speed Packet Switches 749
Mohammad Javad Rostami and Ali Asghar Khodaparast

A New Class of Data Dissemination Algorithms for Multicast Protocols 754
Mohammad Javad Rostami and Ali Asghar Khodaparast

Dynamic Point Coverage in Wireless Sensor Networks: A Learning Automata Approach 758
M. Esnaashari and M.R. Meybodi

A Novel Approach in Adaptive Traffic Prediction in Self-sizing Networks Using Wavelets 763
Hamed Banizaman and Hamid Soltanian-Zadeh

Simulative Study of Two Fusion Methods for Target Tracking in Wireless Sensor Networks 769
Saeid Pashazadeh and Mohsen Sharifi

The Advantage of Implementing Martin's Noise Reduction Algorithm in Critical Bands Using Wavelet Packet Decomposition and Hilbert Transform 773
Milad Omidi, Nima Derakhshan, and Mohammad Hassan Savoji

Real-Time Analysis Process Patterns 777
Naeem Esfahani, Seyed-Hassan Mirian-Hosseinabadi, and Kamyar Rafati

Selecting Informative Genes from Microarray Dataset Using Fuzzy Relational Clustering 782
Soudeh Kasiri-Bidhendi and Saeed Shiry Ghidary

GMTM: A Grid Transaction Management Model 786
Ali A. Safaei, Mostafa S. Haghjoo, and Mohammad Ghalambor

Design of a Custom Packet Switching Engine for Network Applications 790
Mostafa E. Salehi, Sied Mehdi Fakhraie, Abbas Banaiyan, and Abbas Hormati

Multiple Robots Tasks Allocation: An Auction-Based Approach Using Dynamic-Domain RRT 795
Ali Nasri Nazif, Ehsan Iranmanesh, and Ali Mohades

Efficient Computation of N-S Equation with Free Surface Flow Around an ACV on ShirazUCFD Grid 799
Seyyed Mehdi Sheikhalishahi, Davood Alizadehrad, GholamHossein Dastghaibyfard, Mohammad Mehdi Alishahi, and Amir Hossein Nikseresht

Enhancing Accuracy of Source Localization in High Reverberation Environment with Microphone Array 803
Nima Yousefian, Mohsen Rahmani, and Ahmad Akbari

Class Dependent LDA Optimization Using Genetic Algorithm for Robust MFCC Extraction 807
Houman Abbasian, Babak Nasersharif, and Ahmad Akbari

TACtic- A Multi Behavioral Agent for Trading Agent Competition 811
Hassan Khosravi, Mohammad E. Shiri, Hamid Khosravi, Ehsan Iranmanesh, and Alireza Davoodi

Software Reliability Prediction Based on a Formal Requirements Specification 816
Hooshmand Alipour and Ayaz Isazadeh

ID-Based Blind Signature and Proxy Blind Signature without Trusted PKG 821
Yihua Yu, Shihui Zheng, and Yixian Yang

The Combination of CMS with PMC for Improving Robustness of Speech Recognition Systems 825
Hadi Veisi and Hossein Sameti

A Dynamic Multi Agent-Based Approach to Parallelizing Genetic Algorithm 830
Mohsen Momeni and Kamran Zamanifar

Fuzzy Neighborhood Allocation (FNA): A Fuzzy Approach to Improve Near Neighborhood Allocation in DDB 834
Reza Basseda, Maseud Rahgozar, and Caro Lucas

OPWUMP: An Architecture for Online Predicting in WUM-Based Personalization System 838
Mehrdad Jalali, Norwati Mustapha, Md Nasir B. Sulaiman, and Ali Mamat

Artificial Intelligent Controller for a DC Motor 842
Hadi Delavari, Abolzafl Ranjbar Noiey, and Sara Minagar

A Multi-Gb/s Parallel String Matching Engine for Intrusion Detection Systems 847
Vahid Rahmanzadeh and Mohammad Bagher Ghaznavi-Ghoushchi

A Modified Version of Sugeno-Yasukawa Modeler 852
Amir H. Hadad, Tom Gedeon, Saeed Shabazi, and Saeed Bahrami

Directing the Search in the Fast Forward Planning 857
Seyed Ali Akramifar and Gholamreza Ghassem-Sani

A Dynamic Mandatory Access Control Model 862
Jafar Haadi Jafarian, Morteza Amini, and Rasool Jalili

Inferring Trust Using Relation Extraction in Heterogeneous Social Networks 867
Nima Haghpanah, Masoud Akhoondi, and Hassan Abolhassani

Sorting on OTIS-Networks 871
Ehsan Akhgari, Alireza Ziaie, and Mohammad Ghodsi

A Novel Semi-supervised Clustering Algorithm for Finding Clusters of Arbitrary Shapes 876
Mahdieh Soleymani Baghshah and Saeed Bagheri Shouraki

Improving Quality of Voice Conversion Systems 880
M. Farhid and M.A. Tinati

Feature Selection SDA Method in Ensemble Nearest Neighbor Classifier 884
Fateme Alimardani, Reza Boostani, and Ebrahim Ansari

A Novel Hybrid Structure for Clustering 888
Mahdi Yazdian Dehkordi, Reza Boostani, and Mohammad Tahmasebi

6R Robots; How to Guide and Test Them by Vision? 892
Azamossadat Nourbakhsh and Moharram Habibnezhad Korayem

Hierarchical Diagnosis of Vocal Fold Disorders 897
Mansour Nikkhah-Bahrami, Hossein Ahmadi-Noubari, Babak Seyed Aghazadeh, and Hossein Khadivi Heris

A Voronoi-Based Reactive Approach for Mobile Robot Navigation 901
Shahin Mohammadi and Nima Hazar

Evaluation of PersianCAT Agent's Accepting Policy in Continuous Double Auction, Participant in CAT 2007 Competition 905
Sina Honari, Amin Fos-hati, Mojtaba Ebadi, and Maziar Gomrokchi

A Framework for Implementing Virtual Collaborative Networks – Case Study on Automobile Components Production Industry 909
Elham Parvinnia, Raouf Khayami, and Koorush Ziarati

Virtual Collaboration Readiness Measurement a Case Study in the Automobile Industry 913
Koorush Ziarati, Raouf Khayami, Elham Parvinnia, and Ghazal Afroozi Milani

Facilitating XML Query Processing Via Execution Plan 917
Sayyed Kamyar Izadi, Vahid Garakani, and Mostafa S. Haghjoo

The Impact of Hidden Terminal on WMNet Performance 921
Kourosh Hassanli, Ali Khayatzadeh Mahani, and Majid Naderi

Design and Implementation of a Fuzzy Accident Detector 926
Shahram Jafari, Mohammad Arabnejad, and Ali Rashidi Moakhar

Approximation Algorithms for Edge-Covering Problem 930
Mohammad Hosseinzadeh Moghaddam and Alireza Bagheri

A Novel Crosstalk Estimator after Placement 934
Arash Mehdizadeh and Morteza Saheb Zamani

A New Operator for Multi-addition Calculations 938
Kooroush Manochehri, Saadat Pourmozaffari, and Babak Sadeghian

Quantum Differential Evolution Algorithm for Variable Ordering Problem of Binary Decision Diagram 942
Abdesslem Layeb and Djamel-Eddine Saidouni

A Joint Source-Channel Rate-Distortion Optimization Algorithm for H.264 Codec in Wireless Networks 946
Razieh Rasouli, Hamid R. Rabiee, and Mohammad Ghanbari

Pre-synthesis Optimization for Asynchronous Circuits Using Compiler Techniques 951
Sharareh ZamanZadeh, Mehrdad Najibi, and Hossein Pedram

Absolute Priority for a Vehicle in VANET 955
Rostam Shirani, Faramarz Hendessi, Mohammad Ali Montazeri, and Mohammad Sheikh Zefreh

An Ontology Based Routing Index in Unstructured Peer-to-Peer Networks 960
Hoda Mashayekhi, Fatemeh Saremi, Jafar Habibi, Habib Rostami, and Hassan Abolhassani

Two Architectural Practices for Extreme Programming 964
Amir Azim Sharifloo, Amir S. Saffarian, and Fereidoon Shams

Adaptive Target Detection in Sensor Networks 968
Ghasem Mirjalily

Linear Temporal Logic of Constraint Automata 972
Sara Navidpour and Mohammad Izadi

Using Social Annotations for Search Results Clustering 976
Sadegh Aliakbary, Mahdy Khayyamian, and Hassan Abolhassani

Event Detection from News Articles 981
Hassan Sayyadi, Alireza Sahraei, and Hassan Abolhassani

Architectural Styles as a Guide for Software Architecture Reconstruction 985
Kamyar Khodamoradi, Jafar Habibi, and Ali Kamandi

Best Effort Flow Control in Network-on-Chip 990
Mohammad S. Talebi, Fahimeh Jafari, Ahmad Khonsari, and Mohammad H. Yaghmaee

Prevention of Tunneling Attack in endairA 994
Mohammad Fanaei, Ali Fanian, and Mehdi Berenjkoub

Digital Social Network Mining for Topic Discovery 1000
Pooya Moradianzadeh, Maryam Mohi, and Mohsen Sadighi Moshkenani

Towards Dynamic Assignment of Rights and Responsibilities to Agents (Short Version) 1004
Farnaz Derakhshan, Peter McBurney, and Trevor Bench-Capon

Finding Correlation between Protein Protein Interaction Modules Using Semantic Web Techniques 1009
Mehdi Kargar, Shahrouz Moaven, and Hassan Abolhassani

Author Index 1013

A Novel Piecewise Linear Clustering Technique Based on Hyper Plane Adjustment

Mohammad Taheri, Elham Chitsaz, Seraj D. Katebi, and Mansoor Z. Jahromi

{mtaheri,chitsaz}@cse.shirazu.ac.ir,
{katebi,zjahromi}@shirazu.ac.ir

Abstract. In this paper, a novel clustering method is proposed which is done by some hyper planes in the feature space. Training these hyper-planes is performed by adjusting suitable bias and finding a proper direction for their perpendicular vector so as to minimize Mean-Squared Error. For this purpose, combination of training a hyper plane and a fundamental search method named *Mountain-Climbing* is utilized to find a local optimum solution. The approach achieves a satisfactory result in comparison with the well known clustering methods such as k-means, RPCL, and also two hierarchical methods, namely, Single-Link and Complete-Link. Low number of parameters and linear boundaries are only some merits of the proposed approach. In addition, it finds the number of clusters dynamically. Some two dimensional artificial datasets are used to assess and compare these clustering methods visually.

Keywords: Clustering, Unsupervised learning, Pattern Recognition, Linearity, Parameter Adjustment, Search Strategy.

1 Introduction

Different similarity measurement is one of sailent features of clustering methods. Euclidean distance is used in production of hyper spherical clusters, and Mahalanobis distance as a more general approach is utilized in order to form hyper ellipsoidal clusters. But use of these distances (dissimilarity measure) leads to nonlinear clusters. However, some patterns may be distributed and should be discriminated linearly. This linearity may be gained from some linear obstacles in distribution. For example, a special type of soil, which is usually formed in linear layers, may prevent to produce some resources of earthquake.

Although, many instance based clustering methods such as k-means can also produce piecewise linear clusters; but due to its nonlinear approach, it forms correct clusters if real desired discriminator boundaries of each pair of clusters approximately coincide with the perpendicular bisector of associated cluster centers. Also it never can detect clusters which are considerably stretched in a special direction more or less than others; although Single-Link is suitable for these cases. Clusters with different sizes are often hard to be formed by distance based clustering methods.

A statistical approach to the patterns in the feature space without looking at the boundary patterns may be one of the most important drawbacks of some instance based clustering methods such as k-means and RPCL. However, some hierarchical

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 1–8, 2008.

methods such as Single-Link and Complete-Link consider boundary properties more than statistical ones. This boundary based viewpoint may accompany instability; for example, existence, absence or change in position of one pattern may form completely different cluster shapes. In this paper, a clustering method has been proposed which uses hyper planes to distinguish cluster patterns in the feature space such that Least-Mean-Squared Error is locally minimized. Training these planes according to positions of all (especially boundary) patterns and considering LMS error simultaneously results in multifaceted approach to both statistical and boundary properties of patterns.

In the next section, a novel method to train just one parameter of a hyper plane in order to reduce the LMS error is presented. Experimental results are presented in section 3 followed by a conclusion in section 4.

2 Hyper Plane Based Piecewise Linear Clustering System

A *D* dimensional feature space can be divided into two disjoint subspaces by a hyper plane (for example a line or a plane in a 2 or 3 dimensional space, respectively). In the same manner, *p* hyper planes can form many (at most 2^p) convex hyper polygonal subspaces in the feature space such that each produced subspace can be represented by its position relative to the any hyper plane. An elementary subspace is a part of feature space which is bounded by some of existing hyper planes and none of other hyper planes crosses it.

Having a set of *P* hyper planes, $K \le R=2^P$ nonempty subspaces are produced which are considered, in this paper, as different piecewise linear clusters. Presenting the patterns in the clustered feature space, some of clusters, which cover no training pattern, are ignored and others are utilized to categorizing patterns into disjoint groups. After this categorization, it is possible to assess the clustering method by any objective function. In this research, mean of patterns in a group is specified as the center of that cluster and Mean-Squared-Error is attempted to be minimized. Finally, having a predefined number of hyper planes, a learning method is needed to train the hyper planes (adjusting their parameters) in order to gain a clustering structure with the optimal objective function for a specified training dataset.

Assuming a *D* dimensional feature space, a hyper plane is represented by a vector $W = [w_1, w_2, \ldots, w_D]^T$ and a scalar value *B* as the perpendicular vector and the bias of hyper plane, respectively. As shown in (1), each plane divides the feature space into 3 parts, points on the hyper plane and two disjoint subspaces on sides.

$$Plane^j : \left(W^j\right)^T.X = \begin{cases} < B^j & namely, bottom-side \\ = B^j & on\,the\,hyper\ plane \\ > B^j & namely, up-side \end{cases} \quad (1)$$

Each point in this space can be represented by a vector $x=[x_1, x_2, \ldots, x_D]^T$ and its location related to j^{th} hyper plane can be identified by (1). Therefore, any hyper plane is tuned by adjusting *D+1* parameters (one weight per dimension and a bias). Without lose of generality, tuning parameters of hyper planes is sufficient to produce any possible situation in the system where the labels are not important (unsupervised learning). Hence, *P*(D+1)* parameters should be tuned; where, *P* is the number of

hyper planes. In the next section, adjusting a specified parameter is explained whereas values of others are predefined and considered fixed.

2.1 Parameter Adjustment

Here, a simple parameter adjustment is considered which is used in [2, 3] for specifying classification threshold and also in [4, 5] for determining the rule weights in a fuzzy classification system. Both problems are in the field of supervised learning and use classification accuracy based on ROC. In this paper, these techniques are generalized and applied to the linear clustering problem mentioned in the previous section.

Assume V is a parameter in the system under tuning with the object function F. Value of F may change by different values of V, considering other parameters are specified and fixed. In this situation, the objective function can be simplified by substitution of other parameters with their values such that it will be dependent only on the value of V. From now, this simplified version of F is called F^s. If F^s is continuous and analytical, the best value of V can be selected from the finite set of values which make derivative function of F^s equal to zero. But often, this objective function is not analytical. In such situations, another method may be used if the following condition is satisfied:

Value of F^s never changes infinitely by changing V from *'a'* to *'b'*.

Therefore, there are a finite number of values which are possible for F^s; also the range $[a, b]$ can be divided uniquely into some intervals (c^i, c^{i+1}) such that the value of F^s is constant during any interval and differs for two consequent intervals where, $a = c^1 < c^2 < \ldots < c^m = b$. In this paper, c^i is called the *critical value* and is belonged to just one of intervals (c^{i-1}, c^i) or (c^i, c^{i+1}).

In this process, the goal is to find a value for V with the best value (minimum or maximum) for F^s. One or more intervals may exist where any value of them results in the best value for F^s. For this purpose, *m-1* values $d^1 < d^2 < \ldots < d^{m-1}$ are selected as the candidate values of *m-1* intervals such that d^i is the mean of c^i and c^{i+1}. The mean value is chosen since, on the one hand, it is the mean of the whole values in the interval, and on the other hand, there is usually low certainty about the values near the boundaries of an interval. In the classification, reducing probability of overfitting is the most important motivation for choosing the mean value as the candidate of each interval.

In order to find the best values for parameters of hyper planes, assume that there is only one hyper plane and all parameters are valued and considered fixed except that of associated bias. Finding the best value for bias to minimize the MSE may be the main goal. As depicted in Fig. 1, value of bias is in the range of $(-\infty , +\infty)$ and this range is divided into some intervals such that any value in a specified interval leads to the same categorization on training patterns and consequently the same MSE.

As have been depicted, critical values for the bias are completely related to training patterns such that every pattern produces a critical value. If bias was more or less than the critical value, associated pattern would be located in different side of the hyper plane (line in Fig. 1). Hence, the critical value is equal to the needed bias such that the associated pattern is located exactly on the hyper plane. The algorithm for finding the

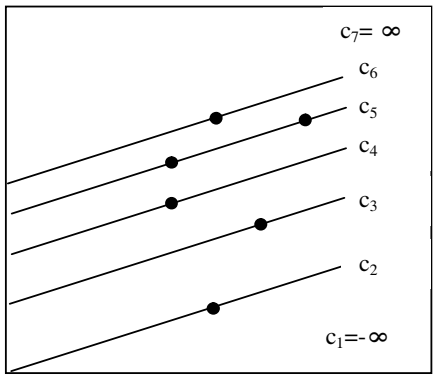


Fig. 1. Critical values for bias of a hyper planebased on the training patterns

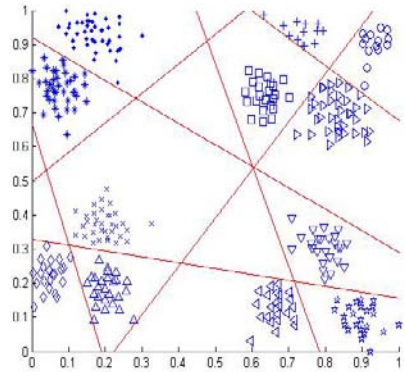


Fig. 2. Proposed method executed on the articial dataset, recognizing 12 clusters by 7 lines

best value for the bias in a hyper plane, when other parameters are fixed, is presented in Table 1.

An alternative is tuning another parameter, for example, without loss of generality, w_j. To do this, we should again compute the critical values, one per training pattern. The same algorithm as Table 1 can be utilized to find the best value of w_j with the following modification:

In Step 2, critical values are computed by (2).

In Step 6, aim is to find the best d^i, as the value of w_j in order to gain the least local Mean-Squared-Error.

$$c_j^i = \left(B - \sum_{1 \le k \le D, k \ne j} \left(x_k^i . w_k \right) \right) / x_j^i \tag{2}$$

Where, B is bias of the only hyper plane considered here in the D dimensional feature space and w_j is the j^{th} element in its perpendicular vector W. Also c_j^i is the critical value for i^{th} pattern x^i in Step 2 when algorithm for finding the best value of w_j is considered. x_j^i is the value of j^{th} feature for x^i.

Table 1. Algorithm of finding the best bias considering other parameters are fixed

STEP 1: LIST OF CRITICAL VALUES $CL = \{\}$

STEP 2: FOR ANY TRAINING PATTERN X^I,
COMPUTE THE ASSOCIATED CRITICAL VALUE C_I
AND INSERT IT INTO THE CL: $c_i = \left(X^i\right)^T . W$

STEP 3: $CL = CL \cup \{\min(CL) - M, \max(CL) + M\}$
WHERE, M IS A BIG NUMBER (100 IN THIS PAPER).

STEP 4: SORT CL AND REMOVE DUPLICATED VALUES.

STEP 5: FOR ANY CONSEQUENT PAIR OF CRITICAL VALUES IN
CL, C_I AND C_{I+1} COMPUTE $D_I = (C_I + C_{I+1})/2$.

STEP 6: FIND THE BEST D_I, TO BE THE BIAS OF THE HYPER PLANE
IN ORDER TO GAIN THE LEAST POSSIBLE MEAN-SQUARED-ERROR

If more than one hype plane exist in the feature space, the same process is repeated to tune a specified parameter of a specified hyper plane considering all other parameters fixed. But in this situation, all hyper planes corporate in dividing the feature space.

2.2 Mountain-Climbing Search Strategy (MCSS)

Mountain-Climbing search is a novel version of Hill-Climbing search [1] with a special dynamic definition for neighborhood. Assuming $V_S = \{V^1, V^2, \ldots, V^q\}$ is the set of parameters such that any solution is represented by a specially assigned values to these parameters. In this method, *q* steps are carried out and in each step a separate parameter is tuned considering values of other parameters fixed. Tuning this parameter should be such that the objective function never decreases. This goal can be achieved if the best value is assigned to each parameter as in the previous section. After tuning all parameters in a predefined order, the iteration can be repeated while no stopping condition is satisfied.

Since, in none of these steps, objective function decreases, the whole process completely tries to reach a solution that tuning no parameter may increases the objective function; but certainly it may be a local optimal solution. Different order of tuning parameters may result in different solutions with different objective function values. Different initialization of parameters can also lead the search to various final solutions. Although this search strategy never guarantees finding the global optimum solution, but our experiments show that its performance is much higher than simple Hill-Climbing. It is not limited to finding the local optimum solution which is probably near to the initial solution. But it can modify a solution by changing only one parameter such that the objective function improves more than expected. Also Mountain-Climbing is stronger than Hill-Climbing in browsing the search space.

2.3 Combination of Hyper Plane Parameter Tuning and MCSS

The proposed method for linearly clustering patterns is based on tuning the parameters of hyper planes by MCSS. The procedure begins with initializing (randomly or not) *P* hyper planes. Afterwards, all hyper planes are considered one by one, and for each hyper plane, *D+1* steps of learning are carried out. In the first *D* steps, *D* parameters of W^r are tuned one by one and in final step B^r is tuned ($1 \le r \le P$). After tuning all parameters of all hyper planes, the iterations can be repeated while no stopping condition is satisfied. Tuning each parameter is carried out, as declared in section 2-1.

In this paper, hyper planes are initiated such that the feature space is divided for each feature into equal parts by perpendicular hyper planes.

2.4 Cluster Punishment

Having *P* hyper planes, $K \le R = 2^P$ different subspaces may be produced at the same time. According to the training patterns, at most *K* nonempty clusters may be produced. Different situations of hyper planes may form different number of

nonempty clusters but there is no control on the real structure of clusters. For example dividing a real cluster of patterns into two or more sub-clusters by one or more hyper planes, is a frequent event in this learning method; since it never considers the number of nonempty clusters and dividing a cluster always reduces the MSE.

To overcome this drawback, a penalty function has been assigned to each nonempty cluster to prevent forming unsuitable clusters. Therefore, objective function *F*, which should be minimized, is modified to (3).

$$F = E_{MSE} + Penalty * NEC \tag{3}$$

Where, E_{MSE} is the *Mean-Squared-Error* and *NEC* is the number of *NonEmpty Clusters* which is multiplied by *Penalty* to be added to error function. But finding the proper value for *Penalty* is a problem. An initial value $Penalty^0$ is assigned to *Penalty* variable and it will be incremented by a constant value *IC* at the end of each iteration of MCSS. $Penalty^0$ and *IC* are both considered equal to 0.001, in our experiments.

The higher value for *Penalty*, the more attention will be paid to the number of clusters. Considering the new objective function proposed by (3), E_{MSE} may increase during the learning process only when the number of clusters is reduced since the error function *F* never increases. Running MCSS with many iterations leads to having high Penalty and consequently results in putting all training patterns in one cluster. Therefore, determination of a good stopping condition is vital. Assuming that F_{MSE}^{i} is the final E_{MSE} at the end of i^{th} iteration of MCSS; the number of iterations ItNo is proposed by (4).

$$ItNo = \max_{i=1,2,\ldots}(i) \ \ where, \ \forall j < k \le i, F_{MSE}^{j} > F_{MSE}^{k} \tag{4}$$

Indeed, MCSS will start a new iteration only if F_{MSE} decreases. After *ItNo* iterations, it is assumed that the value of *Penalty* is suitable for clustering this dataset. With this assumption, and fixing the *Penalty* value, MCSS can be run some (2, in this paper) more iterations in order to tune the parameters further.

2.5 Rotating Points

Assume w_j^i as the j^{th} vector weight of i^{th} hyper plane is to be tuned. Also assume the Z_j^i is the set of points named rotating points, on i^{th} hyper plane such that values of their j^{th} feature are equal to zero. It is considerable that, there is only one point in each set for 2 dimensional feature space and none of these sets are finite for more dimensions. Taking (1) into consideration, changing value of w_j^i, never affects points in Z_j^i and all of them remain on the hyper plane. This may lead to some constraints on the tuning process such that the hyper plane may never find its suitable situation.

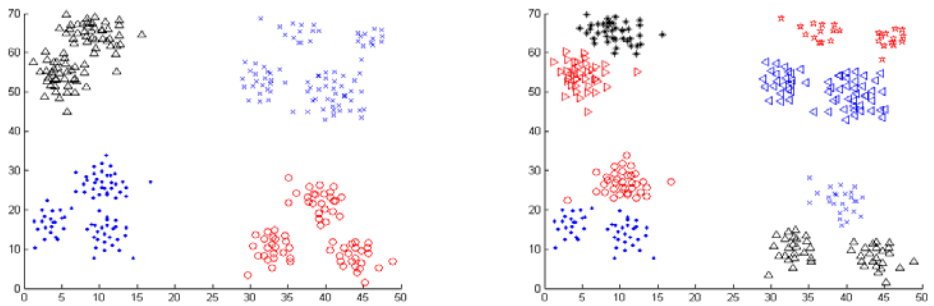
To overcome this drawback, after training the vector weights *W* of a hyper plane, iteratively the origin coordinate is changed to replace the rotating points with new ones related to new origin. After each modification in coordinate of origin, training the vector weights of that hyper plane is done again. Hence, if the origin changes *L* times, each vector weight is tuned *L* times in each iteration of MCSS. Afterwards, the origin is replaced with initial one and bias of that hyper plane is tuned just one time.

Therefore, *(D*L+1)*P* tuning steps are run in each iteration of MCSS where, *D* is the dimension of feature space, *L* is the number of changing origin for each hyper plane in an iteration and *P* is the umber of hyper planes. In Fig. 5, the flowchart for the proposed linear clustering method is depicted and Fig. 6 shows the MCSS for the proposed method.

3 Experimental Results

In this experiment, the performance of the proposed method is compared with ISODATA and agglomerative procedures in clustering an artificial 2-dimentional dataset containing 360 sample data in 4 distinctly disjoint or 12 dense clusters.

Fig. (3-a) shows the result obtained by ISODATA when 4 clusters is specified but it could not split this dataset into 12 clusters, even with many different combination of parameters, unless it acted completely like K-Means. Clearly, the final result for K-Means is highly dependent on initial cluster centers. Fig. (3-b) shows the best result obtained by ISODATA when it is capable of merging or splitting clusters.



(a) **(b)**

Fig. 3. ISODATA executed on the artificial dataset (a) with 4 desired clusters. (b) whereas 12 clusters is desired but only 8 clusters are produced.

Fig. (2) depicts the results obtained by the proposed linear clustering method. The dataset is correctly categorized into exactly 12 clusters without specifying the desired number of clusters in advance. In this experiment 8 initial hyper planes are considered with 3 different origin coordinates, (0, 0), (0.5, 0.5) and (1, 1). As depicted in the Fig. (2), only 7 lines (hyper planes) are remained in the page and other one is biased out of data scope.

4 Conclusion

In this paper, a novel piecewise linear method for clustering based on tuning the parameters of hyper planes is proposed. A method of parameter tuning and a novel search strategy named Mountain-Climbing have been presented which are combined

to train the hyper planes in order to decrease the Mean-Squared-Error of the clusters. Finding the number of clusters dynamically, low number of parameters, linearity and considering statistical and boundary information of patterns at the same time are the main merits for this method. Combination of this method with decision trees, supervised training and use of nonlinear discriminators instead of hyper planes can be considered in the future.

References

1. Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach, 2nd edn., pp. 111–114. Prentice Hall, Upper Saddle River (2003)
2. Fawcett, T.: ROC Graphs: Notes and Practical Considerations for Researchers, Technical Report HPL-2003-4, HP Labs (2003)
3. Lachiche, N., Flach, P.: Improving accuracy and cost of two-class and multi-class probabilistic classifiers using ROC curves. In: Proc. 20th International Conference on Machine Learning (ICML 2003), pp. 416–423 (2003)
4. Zolghadri Jahromi, M., Taheri, M.: A proposed method for learning rule weights in fuzzy rule-based classification systems, Fuzzy Sets and Systems, 29 August (in press, 2007)
5. Zolghadri, M.J., Mansoori, E.G.: Weighting Fuzzy Classification Rules Using Receiver Operating Characteristics (ROC) Analysis. Information Sciences 177(11), 2296–2307 (2007)

Ant Colony Optimization with a Genetic Restart Approach toward Global Optimization

G. Hossein Hajimirsadeghi, Mahdy Nabaee, and Babak N. Araabi

Center of Excellence for Control and Intelligent Processing,
Electrical and Computer Engineering Faculty,
College of Engineering, University of Tehran, Tehran, Iran
{h.hajimirsadeghi,m.nabaee}@ece.ut.ac.ir, araabi@ut.ac.ir

Abstract. Ant Colony Optimization (ACO) a nature-inspired metaheuristic algorithm has been successfully applied in the traveling salesman problem (TSP) and a variety of combinatorial problems. In fact, ACO can effectively fit to discrete optimization problems and exploit pre-knowledge of the problems for a faster convergence. We present an improved version of ACO with a kind of Genetic semi-random-restart to solve Multiplicative Square Problem which is an ill-conditioned NP-hard combinatorial problem and demonstrate its ability to escape from local optimal solutions. The results show that our approach appears more efficient in time and cost than the solitary ACO algorithms.

Keywords: Ant Colony Optimization, NP-hard combinatorial problems, Multiplicative Squares, Heuristic Optimization, Random-Restart Algorithms, Genetic Algorithms, Ill-conditioned Problems.

1 Introduction

Ant algorithms are a class of population-based metaheuristic algorithms for solving Combinatorial Optimization problems. Ant Colony Optimization (ACO) is biologically inspired from the foraging behavior of real ants. ACO is an iterative process in which repeatedly, probabilistic candidate solutions are constructed by heuristic knowledge of the problem and pheromone trails as communication mediums. The main points of ACO are distributed computation, positive feedback and greedy construction heuristics. After the first ACO algorithm proposed by Dorigo (1992) [1], different types of ACO have been developed, most pursuing new ways of exploration and exploitation. Moreover, the combination of ACO and local search algorithms has led to successful results and obtained better performance on variety of problems. To date, ACO has been applied in many combinatorial problems, including Traveling Salesman Problem (TSP), quadratic assignment, vehicle routing, graph coloring, routing for telecommunication networks, sequential ordering, scheduling, data mining, and so on.

In this paper, we introduce an improved version of ACO to maximize the score of Multiplicative Squares (MS). The maximum version of MS is a Square such that sum of the products of its rows, columns, diagonals, and broken diagonals is maximum. It's a complicated problem, because a precision of 20+ digits is needed for the dimensions greater than 10. So a very defiant and crafty algorithm must be applied to

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 9–16, 2008.

the problem. A genetic inspired random-restart is added to the ACO algorithm as a survivor approach to refrain from local maxima.

In section 2, Multiplicative Square is introduced, while optimization algorithms, including ACO and local search methods are described in section 3. Next in section 4, our methodology is explained and the results are summarized in section 5. Finally, conclusions are drawn in section 6.

2 Problem Description of Multiplicative Squares

The Multiplicative Squares are a class of squares filled by the numbers 1 to n^2 that the products of their rows, columns diagonals, and broken diagonals have a special feature. The most well-known type of MS is Kurchan square posed by Rodolfo Kurchan (1989), which is originated from magic square. The maximum product minus the minimum one is as small as possible in a Kurchan Square. However, in the MAX version of MS, sum of the following products is maximum. The score function of a MS of dimension 3 is illustrated in Fig. 1.

Rows: 5*1*8 = 40, 3*9*4 = 108, 7*2*6 = 84
Columns: 5*3*7 = 105, 1*9*2 = 18, 8*4*6 = 192
Diagonals: 5*9*6 = 270, 1*4*7 = 28, 8*3*2 = 48
Anti-diagonals: 8*9*7 = 504, 1*3*6 = 18, 5*4*2 = 40
MAXMS: SF= 40+108+84+105+18+192+270+28+48+504+18+40= 1455
Kurchan MS: SF= 504-18 = 486

5	1	8
3	9	4
7	2	6

Fig. 1. An example of a 3*3 multiplicative square and Score Function (SF) evaluation for MAX MS and Kurchan MS

MS problem is an ill-conditioned NP-hard problem in which a small change of indexes may cause larger errors. Therefore, escaping from local optimums seems to be hard in larger dimensions that more precision and more exploration are needed.

In MAX MS, it can be concluded that the greater numbers should be in the same row, column, diagonal, or broken diagonal, so that the products and at last the score will be greater. We use this feature in our approach that will be described in section 4.

3 Optimization Algorithms

3.1 Ant System

The first ACO algorithm called Ant System applied to Traveling Salesman Problem (TSP) by Dorigo. AS makes up the main framework of other ACO algorithms and is considered as a prototype. In TSP each of *m* artificial ants generates a complete tour by a probabilistic rule (1), which is the probability that ant *k* in city *i* visits city *j*.

$$p_{i,j}^{k} = \begin{cases} \frac{[\tau_{i,j}]^{\alpha}.[\eta_{i,j}]^{\beta}}{\sum_{l \in N_i^k}[\tau_{i,l}]^{\alpha}.[\eta_{i,l}]^{\beta}}, & \forall j \in N_i^k \\ 0, & otherwise \end{cases} \quad (1)$$

Where τ is pheromone, $\eta_{i,j}$ is heuristic function and is equal to $\frac{1}{d_{i,j}}$ the inverse of the difference between city *i* and *j*, N_i^k is the set of cities that haven't been visited by ant *k*, α and β are parameters which shows the relative importance of pheromone versus heuristic or exploitation versus exploration.

Equation (1) shows that ants prefer paths with shorter length and higher amount of pheromone, so they independently generate tours by pre-knowledge of the problem and cooperative informative communication. Once all the ants complete their tours the pheromone trails updates, using (2) and (3).

$$\boldsymbol{\tau_{i,j} = (1-\rho).\,\tau_{i,j} + \sum_{k=1}^{m} \Delta\tau_{i,j}^{k}} \tag{2}$$

$$\boldsymbol{\Delta\tau_{i,j}^{k} = \begin{cases} \frac{Q}{L_k}, & (i,j) \in \text{tour done by ant k} \\ 0, & \text{otherwise} \end{cases}} \tag{3}$$

Where ρ is evaporation rate, L_k is the length of tour taken by ant *k*, *Q* is a constant, and *m* is the number of ants.

3.2 ACO Algorithms

After Ant System, Researchers started to improve the performance of ACO. A first improvement of ACO was elitist strategy (AS_{elite}) [2], which was simply considered more emphasis on the global-best tour. Another improvement was AS_{rank} as an offspring of AS_{elite}, proposed by Bullnheimer, Hartl and Strauss [7]. It sorts the ants and then the trails are updated by only the first $\omega - 1$ ants according to (4).

$$\boldsymbol{\tau_{i,j} = (1-\rho).\,\tau_{i,j} + \sum_{k=1}^{\omega-1}(\omega-1).\,\Delta\tau_{i,j}^{k} + \omega.\,\Delta\tau_{i,j}^{gb}} \tag{4}$$

Where $\Delta\tau_{i,j}^{k} = \frac{1}{L_k}$ and $\Delta\tau_{i,j}^{gb} = \frac{1}{L_{gb}}$.

Stüzle and Hoos introduced MAX-MIN Ant System (MMAS) [8]. In MMAS trails are limited to an interval $[\tau_{min} \ \tau_{max}]$, so it help ants not to converge to local optimum. Further, in MMAS, only the best ant (iteration-best or global-best) is allowed to deposit pheromone. Sometimes, for more exploration an additional mechanism called Pheromone Trail Smoothing is applied to MMAS.

Gambardella and Dorigo in 1996 proposed Ant Colony System (ACS) [11], [3], which was a simplified version of Ant-Q. Ant-Q is a link between reinforcement learning and Ant Colony Optimization. However, ACS simply and more efficiently describes the same behavior as Ant-Q. Two strategies are used in ACS to increase the previous information exploitation. At first, trails are updated by the best ant, like MMAS, and secondly, ants select the next city, using a *pseudo-random proportional rule* [11]. The rule states that with probability q_0 the city *j* is selected, where $j = \arg\max_{j \in N_i^k}\{[\tau_{i,j}]^{\alpha}.[\eta_{i,j}]^{\beta}\}$, while with the probability $1 - q_0$ a city is chosen using (1). Furthermore, there is a distinct difference between ACS and other ACO algorithms and that is trails are updated, while the solutions are built. It's similar to ant-quantity and ant-density approaches that update pheromone trails synchronic to making tours. However, in ACS ants eat portion of the trails as they walk on the path. So the probability that the same solutions are constructed in an iteration decreases.

AS Local Best Tour (AS-LBT) [6], is another improved kind of AS, in which only local information is used to reinforce trails. It means that each ant updates its trail by the best tour it has found to date. This approach shows more diversity than AS.

Some other improvements in the field of ACO are the Multiple Ant Colonies Algorithms [9], which exploits interactions between colonies, Population-based ACO (P-ACO), which makes up a special population of good solutions, and Omicron ACO (OA), which is inspired by MMAS and elitist strategy.

In addition a number of hybrid algorithms have been developed that use good features of ACO. For example the combination of Genetic Algorithm (GA) and ACO, called Genetic Ant Colony Optimization (GACO) have been used to solve different combinatorial problems [4], [5].

Moreover, ACO algorithms often exploit Local Search to improve their performance which is explained in the next section.

3.3 Local Search Algorithms

Local search is a greedy algorithm for solving optimization problems by exploring among candidate solutions. It starts with an initial solution and iteratively moves to neighbor solutions. If a better solution is found it will be replaced by the previous one and the procedure is repeated until no improving solution can be found.

The very simple version of local search is Hill Climbing, in which the first closest neighbor is selected to move. The other well-known local search algorithms are 2-opt and 3-opt.

There exist two crucial problems with local search algorithms that they are easily get trapped in local optimum, and finally their results are strictly dependent on initial solutions [10]. For the first problem some solutions have been devised, like random-restart strategies, while for the latter, heuristic and evolutionary algorithms like ACO can be used to generate appropriate initial solutions for local search algorithms [11].

4 Methodology

In this section we introduce our approach to solve the MAX MS problem. In ACO metaheuristic the main task is to find a graph representation for the problem so that ACO searches for a minimum cost path over the graph.

In each iteration, ants construct a candidate solution individually, going from the first layer to the last one. In Each layer a number between 1 to n^2 is selected which has not been selected with the same ant before. These numbers are used as indices for feasible Multiplicative Squares. It means that the numbers n^2 to 1 are placed in the square respectively, according to indices generated by the ants.

Once the tours are completed ants deposit pheromones on the edges, using (5).

$$\Delta\tau_{i,j}^{k} = \begin{cases} \frac{FS_k}{Q}, & (i,j) \in \textbf{tour done by ant k} \\ \mathbf{0}, & \textbf{otherwise} \end{cases} \tag{5}$$

Where FS_k is sum of the product of rows, columns, diagonals, and broken diagonals (or Function Score) of the square, constructed by ant k. Note that the ratio $\frac{FS_k}{Q}$ is inverse to one suggested in (3) because of the maximization case.

Here we suggest a heuristic function based on the feature, introduced in section 2. The heuristic function applied to each index according to a defined rule. When an ant is in an arbitrary index the heuristic function for the next step to any index in the same row, column, diagonal, or broken diagonal is λ and to other indices is μ which is less than λ. In this respect there is more probability for greater numbers to be multiplied and a larger FS is obtained. For example the mechanism has been shown for a 4 by 4 square in Fig. 2.

μ	λ	λ	λ
λ	λ	*	λ
μ	λ	λ	λ
λ	μ	λ	μ

(a)

λ	μ	λ	λ
λ	λ	λ	*
λ	μ	λ	λ
μ	λ	μ	λ

(b)

Fig. 2. Heuristic function is illustrated for two sample conditions. The current position of the ant is displayed by "*".

For ACO algorithm we used MMAS with a little difference. Actually In our method, the best ants, both iteration-best and global-best deposit pheromone and the heuristic function changes as the iteration increases. Adding iteration-best ants as the communicative elements are for more escape from so many local optimums in the problem, and global-best ants speed up the convergence. Parameter β decreases by the iterations and then increases in the last part of the process for modulating exploration during the search algorithm. Moreover, eating ants are used in the case of local optimum trap which are the components of ACS algorithm.

In the next step, ACO algorithms are accompanied by local search algorithms. In fact pheromone trails are updated by the local search solutions. In our approach the best tours (g-b and i-b) which are obtained in that iteration get improved by 2-opt local search.

As we said in section 3, local search algorithms may get stuck at a local optimum, so we pose a genetic semi-random restart that runs in an outer loop and endows new survivor initial solution to commence ACO and local search algorithms again.

In our genetic restart process, 2 parents reproduce 3 different children by a kind of cross over operator and each of the parents grants a child by mutation.

In cross over two break cuts are selected from 2 to $n^2 - 1$ randomly. Next, the block between these two numbers are chosen from the first parent and then moved to the right corner, left corner, or the same place of a new tour to devote 3 distinct children. The remaining vertices are filled with the other parent. An example of our cross over is depicted in Fig. 3.

In mutation, the block is remained for the parent 2 and the remainder is filled by random permutation. Just the same, the complement block of parent 1 is constant and the remnant is built randomly (Fig. 4). Hence, the two new children are different from the three previous ones, reproduced by cross over as much as possible.

Parent 1	1	**3**	**4**	2	5
Parent 2	4	5	1	2	3
Child 1	3	4	5	1	2
Child 2	5	1	2	3	4
Child 3	5	3	4	1	2

Fig. 3. An example of two cut cross over with 3 children

Parent 1	1	**3**	**4**	2	5
Parent 2	4	**5**	**1**	2	3
Child of parent 1	1	4	3	2	5
Child of parent 2	2	5	1	4	3

Fig. 4. An example of a two cut mutation

5 Experimentation and Results

To verify the efficiency of the proposed algorithm, it was employed on MS7 (7*7 grid) and MS8. Experimentally, We used parameter settings, $\alpha = 1$, initial value of $\beta = 3$, $\rho = .4$, $Q =$ the best SF found up to that iteration [6], eat rate = .9, $\lambda = 1$, and $\mu = .5$ for all experiments. In the case of MS7, the population size of about 50 (equal to the number of variables [2], [12]) ants was used, and the trails were set to interval, [0.002 , 2], with an initial value of 2. While for MS8, population size was set to 64, and trails were limited to $[\tau_{min}, \tau_{max}] = [.001, 2]$ and initial value of 2.

10 trials were conducted, and all the tests were carried out for 600 iteration. The results are presented in Table 1. Flexible heuristic is our complete algorithm with β modulation described in previous section and genetic random restart, while fixed

Table 1. Experiment results

(a) MS7

Method	Best	Avg.	Std. Dev.	Std. Dev %	Best err.%	Avg. err.%
Flexible heuristic	836927418654	836545183884.3	310273380.3	0.037	0	0.046
Fixed heuristic	836864383934	836387896300.2	282729277	0.034	0.0075	0.064
No GA restart	836590536598	835890051299.2	472719981.5	0.057	0.0403	0.124

(b) MS8

Method	Best	Avg.	Std. Dev.	Std. Dev %	Best err.%	Avg. err.%
Flexible heuristic	402702517088866	402397450057731	410397887424.8	0.102	0	0.076
Fixed heuristic	402693316462602	396228893243407	12487304223038.1	3.15	0.0023	1.608
No GA restart	402672245516278	379411679729931	27191910644358.2	7.17	0.0075	5.784

Table 2. Genetic Semi-Random-Restart Performance

Method	Avg. number of successive genetic restart (MS7)	Avg. number of successive genetic restart (MS8)
Fixed heuristic	1.6	2.4
Flexible heuristic	1.3	2.3

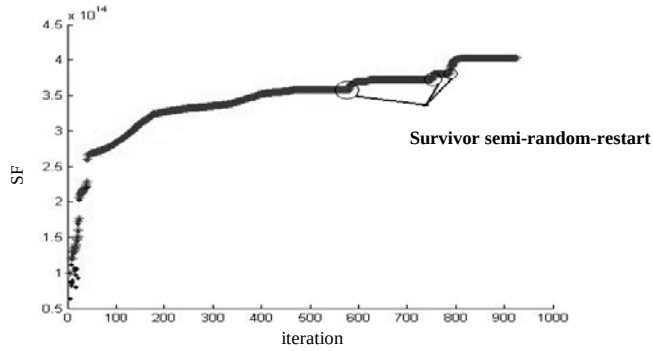


Fig. 5. Successful operation of the posed restart algorithm to evade local optimums

heuristic is the same as the first one without β modulation, and finally, No GA restart represents the algorithm without any restart process.

Table 1 shows a good performance of our algorithm specially compared with the same procedure without restart algorithm. Furthermore, the average number of incidents that the introduced Genetic restart algorithm granted new initial survivor solution to ACO algorithm is stated in Table 2.

To illustrate the operation of our Genetic semi-random-restart algorithm, trace of a particular run is demonstrated in Fig. 5. It shows that the posed restart mechanism improves the robustness and precision of the whole algorithm and efficiently helps to come out of the local optimums.

6 Summary and Conclusion

This paper has introduced an improved version of ACO, with the aid of local search algorithms and specially a genetic restart algorithm, in order to global optimization. Max Multiplicative Square (MS) problem was studied as an ill-conditioned NP-hard combinatorial problem and a particular heuristic function was devised for that. Results have shown that our approach was successful to satisfy the goal of global optimization.

Further work can be in the direction of testing new random-restart techniques, in particular those which substantially differ from local search mechanism. About the MS problem, a better heuristic function can be a great step to decrease the time of evaluation. In addition, a new graph representation might be designed that more efficiently exploit the features of problem, such as symmetry and the importance of big numbers.

References

1. Dorigo, M.: Optimization, Learning and Natural Algorithms (in Italian). PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy (1992)
2. Dorigo, M., Maniezzo, V., Colorni, A.: The Ant System: Optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics – Part B 26, 29–41 (1996)
3. Gambardella, L.M., Dorigo, M.: Solving symmetric and asymmetric TSPs by ant colonies. In: Proceedings of the 1996 IEEE International Conference on Evolutionary Computation (ICEC 1996), pp. 622–627. IEEE Press, Piscataway (1996)
4. Lee, Z.J.: A hybrid algorithm applied to travelling salesman problem. In: 2004 IEEE International Conference on Networking, Sensing and Control, vol. 1, pp. 237–242 (2004)
5. Fu, T.P., Liu, Y.S., Chen, J.H.: Improved Genetic and Ant Colony Optimization Algorithm for Regional Air Defense WTA Problem. In: First International Conference on Innovative Computing, Information and Control (ICICIC 2006), pp. 226–229 (2006)
6. White, T., Kaegi, S., Oda, T.: Revisiting Elitism in Ant Colony Optimization. In: Cantú-Paz, E., Foster, J.A., Deb, K., Davis, L., Roy, R., O'Reilly, U.-M., Beyer, H.-G., Kendall, G., Wilson, S.W., Harman, M., Wegener, J., Dasgupta, D., Potter, M.A., Schultz, A., Dowsland, K.A., Jonoska, N., Miller, J., Standish, R.K. (eds.) GECCO 2003. LNCS, vol. 2723, pp. 122–133. Springer, Heidelberg (2003)
7. Bullnheimer, B., Hartl, R.F., Strauss, C.: A new rank-based version of the Ant System: A computational study. Central European Journal for Operations Research and Economics 7, 25–38 (1999)
8. Stüzle, T., Hoos, H.: The MAX–MIN Ant System and local search for the traveling salesman problem. In: IEEE International Conference on Evolutionary Computation (ICEC 1997), pp. 309–314. IEEE Press, Piscataway (1997)
9. Kawamura, H., Yamamoto, M., Suzuki, K., Ohuchi, A.: Multiple Ant Colonies Algorithm Based on Colony Level Interactions. IEICE Transactions E83-A, 371–379 (2000)
10. Dorigo, M., Stüzle, T.: The Ant Colony Optimization Metaheuristic: Algorithms, Application and Advances. Technical Report, IRIDIA-2000-32 (2000)
11. Dorigo, M., Gambardella, L.M.: Ant Colony System: A cooperative learning approach to the traveling salesman problem. IEEE Transactions on Evolutionary Computation 1, 53–66 (1997)
12. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence From Natural to Artificial Systems. Oxford University Press, New York (1999)

Automatic Extraction of IS-A Relations in Taxonomy Learning

Mahmood Neshati, Hassan Abolhassani, and Hassan Fatemi

Web Intelligence Laboratory
Computer Engineering Department
Sharif University of Technology, Tehran, Iran
{neshati,fatemi}@ce.sharif.edu, abolhassani@sharif.edu

Abstract. Taxonomy learning is a prerequisite step for ontology learning. In order to create a taxonomy, first of all, existing 'is-a' relations between words should be extracted. A known way to extract 'is-a' relations is finding lexico-syntactic patterns in large text corpus. Although this approach produces results with high precision but it suffers from low values of recall. Furthermore developing a comprehensive set of patterns is tedious and cumbersome. In this paper, firstly, we introduce an approach for developing lexico-syntactic patterns automatically using the snippets of search engine results and then, challenge the law recall of this approach using a combined model, which is based on co-occurrence of pair words in the web and neural network classifier. Using our approach both precision and recall of extracted 'is-a' relations improved and F-Measure value reaches 0.72.

Keywords: Semantic Web, Ontology Engineering, Taxonomy Learning.

1 Introduction

The final goal of semantic web is creating structures with machine processable information. These structures are the basis of machine processing. Extending practical usage of semantic web is based on development of these structures. Ontology is the most known example of such a structure. Hence, developing automatic or even semi-automatic approaches to ease the process of ontology construction is known as a basic factor in developing semantic web.

Taxonomy extraction from concepts of a domain is the prerequisite phase for constructing ontology for that domain. In fact, taxonomy constitutes the backbone of ontology, and its quality directly affects the quality of the constructed ontology.

The formal definition of taxonomy borrowed from [1] is as follows:

Taxonomy is a triplet $T = (C, root, \leq_C)$ where:

C, a set of concepts. By concepts we mean words of a specific domain. The root node that represents top element.

A partial order relation $(\leq c)$ defined on elements $C \cup \{root\}$ is defined as: $\forall c \in C: c \leq c$ root. If $c_1 \leq c\, c_2$ means that concept $c_1$has 'is-a' relation with ceptc_2. Such as $Man \leq c\, Person$.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 17–24, 2008.

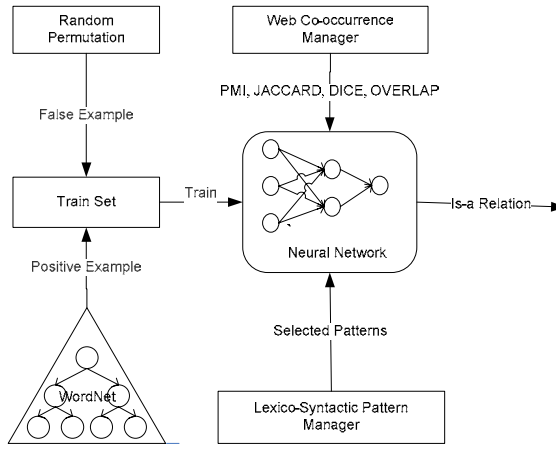


Fig. 1. Compound model for Extracting IS_A Relations

A classic way for finding 'is-a' relations between words is by searching lexico-syntactic patterns in large text corpus. Although this approach results in high values of precision but it doesn't yield promising values of recall.

In this paper we use the web, the most comprehensive text corpus, in order to improve the low values of recall. Therefore, we use the search engines for searching web pages.

Two information elements provided by search engines are the number of pages retrieved for the query Q and an abstraction of the document that contains the concept. In this paper we use these two elements for extracting 'is-a' relations.

Another group of methods for extracting 'is-a' relations is based on distribution hypothesis [2]. In comparison with pattern based methods, this approach has better recall but lower precision value. One of the important attributes of a concept used in computing semantic similarity of two concepts is the number of documents that contains both concepts.

Using search engines we search '$w_1\ and\ w_2$' for finding documents containing both $w_1\ and\ w_2$ words. The documents retrieved for this query represents a good estimation of co-occurrence of words in the web.

Snippet is part of retrieved document surrounding the required concept. It provides valuable information about occurrences of words under different situations in web documents. In this paper we use the snippets for constructing and then searching lexico-syntactic patterns.

In this paper we use a compound model for extracting 'is-a' relations. In addition to co-occurrence of words in web, this model considers lexico-syntactic measures in the process of extracting 'is-a' relations. According to Fig. **1**, we use neural network model to combine the two mentioned methods.

2 Related Works

2.1 IS-A Relation Extraction Methods

Existing methods about extracting 'is-a' relations can be classified into two main groups.

2.1.1 Clustering Based Methods

This group includes methods based on distribution hypothesis [2]. Simply, these methods try to extract a set of attributes. As alluded to it before, according to distribution hypothesis, the more the common attributes of two concepts, the similarity of the concepts and therefore the probability of existence of a 'is-a' relation between them increases. According to diversity of attributes, different similarity measures could be used. For example [3] uses co-occurrence of two words in a window using point wise mutual exclusion(PMI) to compute the similarity value of two words. In [4], different statistical methods have been used to extract attributes and compute the similarity of pair words.

2.1.2 Pattern-Based Methods

Another group of methods for extracting taxonomy includes methods based on lexico-syntactic patterns. Knowledge is represented in different levels of speciosity in documents. The probability of occurrence patterns like the ones shown in Table 1 in general texts are so rare. Therefore many pair words that have 'is-a' relation with each other cannot be recognized by this method. The idea of extracting semantic relations such as 'is part of' and 'casual relation' is applied in [6,7].

Table 1. Hearst Patterns

Example	Pattern
Vehicles such as cars	NP(s) such as NP
such cars as cabriolets	Such NP(s) as NP
motor-bikes and other two-wheeled vehicles	NP (and|or) other Np(s)

3 Proposed Algorithm

3.1 Summery

Using neural network model we combine co-occurrence and pattern-based measures. Section 3.4 explains this combination in more detail. We present a method to extract patterns that 'is-a' relation can be deduced from them automatically from web documents. After extracting the patterns, they must be ranked somehow, so that the best pattern receives the value 1 as its rank. Finding 'is-a' relations can be turn into a classification problem. For a pair of words such as w_1 andw_2, our algorithm must determine whether a 'is-a' relation can be between them or not. In other words is the relation $is-a(w_1, w_2)$ true or not?

In order to solve the classification problem, after extracting and ranking the patterns, we find an optimal combination of co-occurrence in the web and lexico-syntactic patterns factors using a neural network model. To learn suitable patterns we use WordNet for extracting positive samples (words with 'is-a' relations) and negative ones are constructed by random combination of words.

3.2 Co-Occurrence Measure

For computing the co-occurrence rate of two words, such as w_1 and w_2, we search 'w_1 *and* w_2' query using Google search engine. The number of retrieved pages is an estimation of the number of pages that contain both w_1 and w_2. Numerous measures can be developed for computing the co-occurrence of two words.
We used 4 measures for computing co-occurrence of two words which are Jaccard, Dice, Overlap and PMI.

3.3 Lexico-Syntactic Patterns and Snippets

For extracting contextual information from the web, we use snippets of search engine results. There are many suitable patterns for extracting 'is-a' relations that are neglected by manual methods. For example we can refer to the patterns listed inTable 3. These patterns are extracted from snippets automatically.

For every pair of words(w_1, w_2), we search a query and use its snippet set and in each snippet we replace w_1 and w_2 with X and Y respectively. Then, we move a window on each snippet that its size is 10. We consider those windows that have at least one X and one Y as valid windows and finally we count the repeat number of each pattern. We earn 95351 different patterns with this approach. For reducing the number of patterns, we eliminate those that have another pattern as their sub-set. Doing so, only 166 patterns remain and Only 41 patterns of them are repeated more than 5 times in the snippets. In other words only %75 of the patterns happen less than 6 times. Training the neural network is not possible with such a disperse data. Hence, we must pick up appropriate patterns somehow and train the neural network only with those patterns. For determining the amount of power a pattern has in extracting 'is' relation, we use statistical test X^2.

$$\chi^2(v) = \frac{(P+N)\left(p_v(N-n_v)-n_v(P-p_v)\right)^2}{PN(p_v+n_v)(P+N-p_v-n_v)} \quad (1)$$

We need two sets to do that. The first set includes positive samples (words with 'is' relation) and the second one includes negative samples (words without 'is' relation). For each pattern we construct a dependency table like Table 2, in which v is an extracted pattern, p_v is the number of times that v happens in the positive set and n_v is the number of times that v happens in the negative set. Also P and V are the total number of times that patterns happen in the positive set and negative set respectively. For each pattern the X^2 measure is calculated according to the below formula.

For each pattern, the value of X^2shows the suitability scale of that pattern in extracting 'is' relations. Patterns with the highest rank are shown in Table 3.

Table 2. Contingency Table

	v	Other than v	All
Freq. in Positive set	p_v	$P - p_v$	P
Freq. in Negative set	n_v	$N - n_v$	N

Table 3. Sorted Patterns Extracted from Snippets

χ^2	Pattern	χ^2	Pattern	χ^2	Pattern
13.32	X/Y	21.03	Y X	43.14	X Y
9.68	X is a Y	16.15	X of Y	38.20	X and Y
8.46	X is a kind of Y	15.76	X or Y	25.85	Y and X
8.46	Y & X	13.96	Y of X	21.86	X- a Y

3.4 Combining Patterns and Co-Occurrence

As said before, the problem of determining whether there is an 'is-a' relation between two words or not, can be turned into a classifier problem. Having do that, for a pair of words we set the value of target variable to 1 if there would be an 'is-a' relation between them, otherwise we set it to 0. For every pair of words(w_1, w_2), we make a vector $\vec{F}$. Vector $\vec{F}$ has $N + 4$ dimensions and N is the number of lexico-analysis patterns which are used for training the neural network and the other 4 dimension are for co-occurrence measures. Using extracted vectors, we train the neural network and doing that needs positive and negative samples. We use WordNet for constructing the positive samples and consider only those samples as positive that are connected with an edge in WordNet. Negative samples are constructed using random combination of pair words that there is not an 'is' relation between them.

Using the neural network model we can estimate the target variable. In fact, if estimated target variable using our approach is set to 1 it means that there is an 'is-a' relation between them and otherwise there is no 'is-a' relation between them.

4 Experiments

4.1 Designing Test Collection

For constructing a test set we use 200 pair words. 100 pairs of them have hyponym/hypernym relation in WordNet and the other 100 pairs have 'is' relation.

4.2 Performance Evaluation

We use different measures to evaluate the efficiency of our algorithm. For evaluating the efficiency of the classifier we use confusion matrix as shown in Table 4.

An important measure for evaluating the efficiency of the algorithm is precision. We define the precision measure as below. It shows the portion of correctly expected 'is' relations.

Table 4. Confusion Matrix

Total	1(Predicated)	0(Predicated)	Is-a Relation
$\text{Negative}_{\text{Actual}}$	$\text{False}_{\text{Positive}}$	$\text{True}_{\text{Negative}}$	0(Actual)
$\text{Positive}_{\text{Actual}}$	$\text{True}_{\text{Positive}}$	$\text{False}_{\text{Negative}}$	1(Actual)
	$\text{Positive}_{\text{Predicted}}$	$\text{Negative}_{\text{Predicted}}$	Total

$$\textbf{Precision} = \frac{\textbf{True}_{\textbf{Positive}}}{\textbf{False}_{\textbf{Positive}} + \textbf{True}_{\textbf{Positive}}} \tag{2}$$

In addition to precision, recall is also an important measure. This measure shows the portion of pair words with Hyponym/Hypernym relation recognized by that neural network model.

$$\textbf{Recall} = \frac{\textbf{True}_{\textbf{Positive}}}{\textbf{True}_{\textbf{Positive}} + \textbf{False}_{\textbf{Negative}}} \tag{3}$$

In addition to the above two measures, Overall Error Rate shows the error rate of the algorithm.

$$\textbf{Overal Error Rate} = \frac{\textbf{False}_{\textbf{Positive}} + \textbf{False}_{\textbf{Negative}}}{\textbf{Negative}_{\textbf{Actual}} + \textbf{Positive}_{\textbf{Actual}}} \tag{4}$$

There is a trade-off between precision and recall. Increasing one of them decreases the other one. Hence, we use F-Measure as a general measure for evaluating the efficiency of the algorithm.

$$\textbf{F} = \frac{\textbf{2} * \textbf{Precision} * \textbf{Recall}}{\textbf{Precision} + \textbf{Recall}} \tag{5}$$

4.3 Selecting the Optimal Number of Patterns

As said in section 3-3, we extract 166 lexico-syntactic patterns for finding ‘is’ relations and prioritize them according to their power in showing ‘is’ relation. In order to determine the optimal number of patterns for training the neural network, we train it with different number of patterns. In fact, in each experiment the neural network is trained with the N highest rank patterns. Efficiency measures for the algorithm are shown in Fig 2.

It is obvious from Fig 2 that increasing the number of patterns will decrease the precision of the algorithm. It is because that we prioritize patterns according to their ability in deducing ‘is’ relation. It is obvious that increasing the number of patterns will cause weaker patterns to be added to the neural network model, therefore decreasing the precision of the algorithm. In contrast with precision, recall has an increasing behavior because increasing the number of patterns will cause more ‘is’ relations to be extracted and therefore increase the recall.

4.4 Precision Diagram vs. Number of Snippets

We are not able to investigate all retrieved snippets; instead we only investigate those with top rank for extracting patterns. In Fig 3 the efficiency of the algorithm in compare with the number of snippets for each pair words is shown.

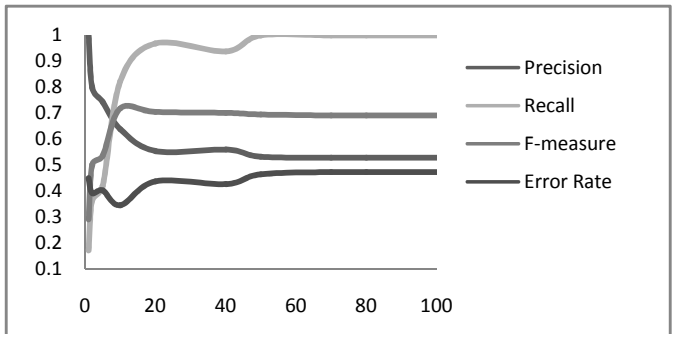


Fig 2. F-measure vs. Number of Patterns

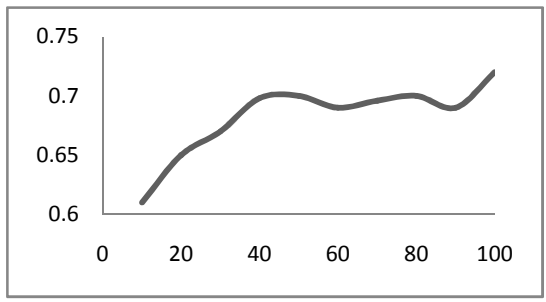


Fig 3. F-measure vs. Number of Snippets

As shown in Fig 3, increasing loaded snippets will increase F-Measure. It is because that, increasing snippets, enables us to find more patterns in them and it also alleviate the problem of data sparseness. It is obvious that increasing the number of snippets will increase the load and process time.

5 Conclusion

According to our experiments, in addition to lexico-syntactic patterns, PMI measure can also play a prominent role in extraction of 'is' relations. In fact, the role of PMI measure in extraction of 'is' relations is increasing recall value and the role of lexico-syntactic patterns is increasing precision value. Hence, we can increase both precision and recall values with special combinations of these measures. Our proposed method, not only present a higher precision value in compare with [5] (0.64 in compare with 0.57), but it also produces higher values of recall by combining co-occurrence measures and lexico-syntactic patterns. Previous works have not report their recall values, so we are not able to compare our recall values with the recall values of them. In this paper we propose an algorithm for extracting 'is' relations using search engines. Our proposed algorithm is based on lexico-syntactic patterns and, on the other hand, it uses co-occurrences of words for extracting 'is' relations. These two measures are

combined by neural network model to create a powerful classifier for deducing 'is' relations. We use a training set, which is produced automatically from WordNet, for training the neural network. The proposed algorithm produces higher values of precision in compare with existing approaches that use limited and non-automatic patterns. Training and execution time complexity is not high, because instead of loading the whole page we only load snippets. We construct an optimal set of lexico-analysis patterns by investigating the impact that the number of patterns has on the efficiency of the algorithm. With this optimal set of patterns the algorithm produces results with 0.64 and 0.82 for precision and recall respectively. The error rate of our proposed algorithm is 0.34. Finally, we used sensitivity analysis for finding the relative importance of different factors that play role in deducing 'is' relations.

Acknowledgement

The research work done in this paper is financially supported by the Iranian Telecommunication Research Center (ITRC) under the research protocol, T/500/1895, on May 7 2007.

References

1. Cimiano, P., Hotho, A., Staab, S.: Learning Concept Hierarchies from Text Corpora using Formal Concept Analysis. Journal of Artificial Intelligence Research 24 (2005)
2. Harris, Z.: Mathematical Structures of Language (1968)
3. Church, K.W., Hanks, P.: Word association norms, mutual information, and Lexicography. In: Association for Computational Linguistics. Proceedings of the 27th Ainguistics, pp. 76–83 (1989)
4. Terra, E., Clarke, C.L.: Frequency Estimates for Statistical Word Similarity Measures. In: Edmonton, A.A. (ed.) Proceedings of Human Language Technology conference 2003, pp. 244–251 (2003)
5. Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. In: 4th International Conference on Computational Linguistics, pp. 539–545 (1992)
6. Berland, M., Charniak, E.: Finding parts in very large corpora. In: 37th Annual Meeting of the ACL, pp. 57–64 (1999)
7. Girju, R., Moldovan, D.I.: Text mining for causal relations. In: Proceedings of the Fifteenth International Florida Artificial Intelligence Research Society Conference, pp. 360–364 (2002)

A Bayesian Network Based Approach for Data Classification Using Structural Learning

A.R. Khanteymoori[1], M.M. Homayounpour[2], and M.B. Menhaj[3]

[1] PhD Candidate, Computer Engineering Department
[2] Assistant Professors, Computer Engineering Department
[3] Professor, Electrical Engineering Department,
AmirKabir University, Tehran, Iran
{khanteymoori,homayoun,menhaj}@aut.ac.ir

Abstract. This paper describes the theory and implementation of Bayesian networks in the context of data classification. Bayesian networks provide a very general and yet effective graphical language for factoring joint probability distributions which in turn make them very popular for classification. Finding the optimal structure of Bayesian networks from data has been shown to be *NP*-hard. In this paper score-based algorithms such as K2, Hill Climbing, Iterative Hill Climbing and simulated annealing have been developed to provide more efficient structure learning through more investigation on MDL, BIC and AIC scores borrowed from information theory. Our experimental results show that the BIC score is the best one though it is very time consuming. Bayesian naive classifier is the simplest Bayesian network with known structure for data classification. For the purpose of comparison, we considered several cases and applied general Bayesian networks along with this classifier to these cases. The simulation results approved that using structural learning in order to find Bayesian networks structure improves the classification accuracy. Indeed it was shown that the Iterative Hill Climbing is the most appropriate search algorithm and K2 is the simplest one with the least time complexity.

Keywords: Bayesian Networks, Data Classification, Machine learning, Structural learning.

1 Introduction

A Bayesian network (BN) [1] consists of a directed acyclic graph G and a set P of probability distributions, where nodes and arcs in G represent random variables and direct correlations between variables respectively, and P is the set of local distributions for each node. Learning BNs has become an active research in the past decade [2]. The goal of learning a BN is to determine both the structure of the network (structure learning) and the set of conditional probability tables CPTs (parameter learning). Learning the structure, or causal dependencies, of a graphical model of probability such as a Bayesian network is often a first step in reasoning under uncertainty. The structure of a Bayesian network encodes variable independencies. Since

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 25–32, 2008.

the number of possible structures is extremely huge, structure learning often has high computational complexity. Finding the optimal structure of a BN from data has been shown to be *NP*-hard [3]. Thus, heuristic and approximate learning algorithms are the realistic solution. Therefore, greedy score-based algorithms [3] have been developed to provide more efficient structure learning at an accuracy tradeoff. This paper aims to propose the use of structural learning in order to determine structure of the BN classifier. K2, Hill Climbing, Iterative Hill climbing and simulated annealing have been developed to provide more efficient structure learning through more investigation on MDL, BIC and AIC scores. The rest of the paper is organized as follows. In Section 2, we discuss about structural learning in Bayesian networks. Next, in Section 3, we describe the naive Bayes classifier. In section 4 we will illustrate implementation results. Finally, Section 5 summarizes the main conclusions.

2 Structural Learning in Bayesian Networks

A Bayesian network is a directed acyclic graph. Directed acyclic graph, also called a DAG, is a directed graph with no directed cycles. Bayesian network provides a compact representation or factorization of the joint probability distribution for a group of variables [4]. In this way, the joint probability distribution for the entire network can be specified. This relationship can be captured mathematically using the chain rule in Equation 1 [6].

$$p(x) = \prod_{i=1}^{n} p(x_i \mid parents(x_i)) \tag{1}$$

We are interested in learning BN from training data D consisting of examples $\mathbf{x}$. The two major tasks in learning a BN are: learning the graphical structure, and then learning the parameters (CP table entries) for that structure. The structure of a BN, defines the conditional independencies among the variables. There are different ways of establishing the Bayesian network structure [9, 4]. Learning structure means learning conditional independencies from observations. The parameters of a BN with a given structure are estimated by ML or MAP estimation. One can use a similar method for choosing a structure for the BN that fits to the observed data. However, structural learning is slightly different than parameter learning. If we consider ML estimation or MAP estimation with uninformative priors then the structure that maximizes the likelihood will be the result. In this case complex structures (i.e. that has more dependencies between variables) will be more capable to increase the likelihood because they have more degree of freedom. However, the aim is to find a structure that encodes the independencies among the variables, that is we want our structure to be as simple as possible.

The common approach to this problem is to introduce a scoring function that evaluates each network with respect to the training data, and then to search for the best network. In general, this optimization problem is intractable, yet for certain restricted classes of networks there are efficient algorithms requiring polynomial time (in the number of variables in the network).

2.1 Scoring Structure

The criteria for selecting a structure have two terms. One for maximizing the likelihood and one for minimizing the complexity. The log-likelihood score is a simple score. While log-likelihood score is very simple, is not suitable for learning the structure of the network, since it tends to favor complete graph structures (in which every variable is connected to every other variable). In order to avoid overfitting we can add a penalty for the complexity of the network based on the number of parameters. One of the important properties of MDL methods is that they provide a natural safeguard against overfitting, because it implements a tradeoff between the complexity of the hypothesis (model class) and the complexity of the data given the hypothesis. MDL says that, the best model of a collection of data is the model that minimizes the length of the encoding. Hence learning a BN based on the MDL score is equivalent to minimizing the penalty score for the complexity of the structure minimizing the likelihood of data given the structure, based on ML parameter estimates [8].

A third possibility is to assign a prior distribution over network structures and find the most likely network by combining its prior probability with the probability accorded to the network by the data. This is the "Bayesian" approach to network scoring. Depending on the prior distribution used, it can take various forms. However, true Bayesians would average over all possible network structures rather than singling out a particular network for prediction. Unfortunately, this generally requires a great deal of computation. BIC score results from the approximation of the likelihood of the data given the structure around the MAP or ML estimates of the parameters, the approximation results in the same formulation of MDL [8]. Akaike's information criterion (AIC) is a measure of the goodness of fit of an estimated statistical model. It is grounded in the concept of entropy. The AIC is an operational way of trading off the complexity of an estimated model against how well the model fits the data.

2.2 Searching Structure Space

Searching the structure space for high scored structures is the next issue in structural learning. It has been shown [4] that finding the structure with maximum scoring is NP-hard. Therefore for arbitrary structures, heuristic search algorithms are used. Bayesian network structural learning algorithms differ mainly in the way in which they search through the space of network structures. Some algorithms are introduced below.

2.2.1 K2 Search

K2 is a greedy algorithm and is initialized with an ordering of nodes such that the parents of a node are listed above the node (i. e. topological ordering). Starting with an empty structure and tracing the node list in order, one adds a parent to the node that increases the score. The number of parents to be added to a single variable may be limited to a predetermined constant for fast inference [9]. K2 finds structures quickly if given a reasonable ordering α. If the number of parents per variable is constrained to a constant upper bound, K2 has worst-case polynomial running time in the number n of variables. Two clear limitations of greediness are inability to backtrack (i.e., undo the addition of an arc) or consider the joint effects of adding multiple arcs (parents). This is why greedy structure learning algorithms are sensitive to the presence of irrelevant variables in the training data, a pervasive problem in machine

learning [9]. Additionally, K2 is particularly sensitive to the variable ordering because arcs fail to be added, resulting in unexplained correlations, whenever candidate parents are evaluated in any order that precludes a causal dependency.

2.2.2 Hill-Climbing

The idea of a hill-climbing search algorithm is to generate a model in a step-by-step fashion by making the maximum possible improvement in an objective quality function at each step.

Initialize with a network structure, possibly random, evaluate the change in the score for all arc changes on this network and choose the one that has the maximum change. Continue this process until no more arc changes increase the score. This algorithm generally sticks into local maxima.

2.2.3 Iterated Hill-Climbing

The problem of getting stuck in local optima is a big drawback of the hill-climbing algorithm. Various other optimization techniques, such as iterated hill-climbing try to overcome this problem. Iterated hill-climbing apply local search until local maximum. Randomly perturb the structure and repeat the process for some manageable number of iterations.

2.2.4 Simulated Annealing

In contrast to the hill-climbing search, simulated annealing allows occasional uphill jumps, allowing the system to hop out of local minima and giving it a better chance of finding the global minimum. Simulated annealing exploits an analogy between the way in which a metal cools and freezes into a minimum energy crystalline structure (the annealing process) and the search for a global optimum in an optimization problem. Similarly to hill-climbing search, simulated annealing employs a random search through the space of possible structures, with the distinction that it accepts not only changes that improve the structure, but also some changes that decrease it. The latter are accepted with a probability $e^{-\frac{\Delta}{T}}$. The pseudo-temperature is gradually lowered throughout the algorithm from a sufficiently high starting value.

3 Bayesian Network Classifier

The naive Bayes classifier is the simplest BN classifier assumes that all the attributes are conditionally independent given the class label. As reported in the literature [10], the performance of the naive Bayes classifier is surprisingly good even if the independence assumption between attributes is unrealistic in most of the data sets. Independence between the features ignores any correlation among them. The Naïve Bayes classifier is a network with an edge leading from the class attribute to each of the other attributes and extending its structure to explicitly represent variable dependencies is a direct way to overcome the limitation of naive Bayes.

When building networks for classification, it sometimes helps to use this network as a starting point for the search. Now that we have the means to represent and manipulate independences, the obvious question follows: Can we learn an unrestricted

Bayesian network from the data that when used as a classifier maximizes the prediction rate?

Using the methods just described we can induce a Bayesian network from the data and then use the resulting model as a classifier. The learned network represents an approximation to the probability distribution governing the domain (given the assumption that the instances are independently sampled form a single distribution). Given enough samples, this will be a close approximation. Thus, we can use this network to compute the probability of C given the values of the attributes. The predicted class c, given a set of attributes $a_1, a_2, ..., a_n$ is simply the class that attains the maximum posterior $P_B(c \mid a_1, a_2, ..., a_n)$, where P_B is the probability distribution represented by the Bayesian network B.

It is important to note that this procedure is unsupervised in the sense that the learning procedure does not distinguish the class variable from other attributes. Thus, we do not inform the procedure that the evaluation of the learned network will be done by measuring the predictive accuracy with respect to the class variable.

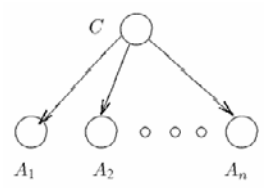


Fig. 1. The structure of the naive Bayes network

4 Experimental Results

We run our experiments on the 5 datasets listed in Table 1.

Table 1. Experimental datasets

	Dataset	Number of instances	Number of attributes	Number of classes
Lens	D 1	24	5	3
iris	D 2	150	4	3
labor	D 3	57	17	3
segment	D 4	1500	20	7
soybean	D 5	683	36	19

The contact lens data (dataset D1) tells us the kind of contact lens to prescribe, given certain information about a patient. The iris dataset (D2), is arguably the most famous dataset used in data mining, contains 150 examples each of three types of plant: Iris setosa, Iris versicolor, and Iris virginica. There are four attributes: sepal length, sepal width, petal length, and petal width (all measured in centimeters). Unlike previous dataset, all attributes have values that are numeric. The labor negotiations

dataset (D3) summarizes the outcome of Canadian contract negotiations in 1987 and 1988. It includes all collective agreements reached in the business and personal services sector for organizations with at least 500 members (teachers, nurses, university staff, police, etc.). Each case concerns one contract, and the outcome is whether the contract is deemed acceptable or unacceptable. The segment dataset (D4) includes image data segmented into classes such as grass, sky, foliage, brick, and cement based on attributes giving average intensity, hue, size, position, and various simple textural features. An often-quoted early success story in the application of machine learning to practical problems is the identification of rules for diagnosing soybean diseases. The data (D5) is taken from questionnaires describing plant diseases. There are 683 examples, each representing a diseased plant. Plants were measured on 35 attributes, each one having a small set of possible values. There are 19 disease categories.

Since the amount of data for training and testing is limited. Cross validation method used which reserves a certain amount for testing and uses the remainder for training. In cross-validation, we decide on a fixed number of folds, or partitions of the data. Then the data is split into n approximately equal partitions and each in turn is used for testing and the remainder is used for training and repeat the procedure n times so that, in the end, every instance has been used exactly once for testing. This is called n-fold cross-validation.

In our experiments we used 10-fold cross-validation. Each part is held out in turn and the learning scheme trained on the remaining nine-tenths; then its error rate is calculated on the holdout set. Thus the learning procedure is executed a total of 10 times on different training. Finally, the 10 error estimates are averaged to yield an overall error estimate. We observed classification performance for each algorithm. The accuracy of each classifier is based on the percentage of successful predictions on the test sets of each dataset. In Figure 2, the accuracies of the four learning procedures discussed in this paper are summarized. Computations were done on a personal computer with a 3.20 GHz Pentium CPU, and 1GB of RAM memory. We used three score measures including BIC, MDL and BIC.

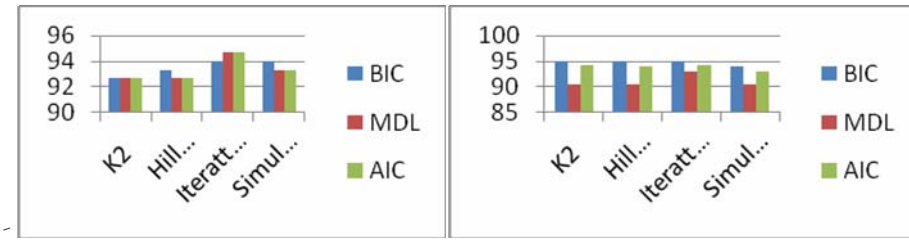


Fig. 2. Clustering Performance on D2 and D4 dataset

Table 2 and Table 3, present the total learning time versus search algorithm and scoring measures when BN classifier is used for classification of data in D4 and D5 datasets. D4 and D5 are selected since they include more instances than other datasets presented in Table 1. As it can be seen, the BIC as a ,scoring measure and Simulated Annealing as a learning algorithm are the most time consuming among the studied measures and algorithms.

Table 2. Computation time for model building (D4 Dataset) in seconds

	K2	Hill Climbing	Iterative Hill Climbing	Simulated Annealing
BIC	8.59	15.28	314.98	878.29
MDL	0.25	0.41	17.82	49.16
AIC	0.31	0.59	23.12	63.73

Table 3. Computation time for model building (D5 dataset) in seconds

	K2	Hill Climbing	Iterative Hill Climbing	Simulated Annealing
BIC	15.41	43.52	745.78	2648.15
MDL	0.08	1.27	32.53	139.21
AIC	0.14	1.49	41.17	154.73

We compared Bayesian network approach which can learn its structure from data with naive Bayes classifier introduced before. As you can see in Figure 3, the results show that the Bayesian network with structural learning outperforms the naïve Bayes classifier.

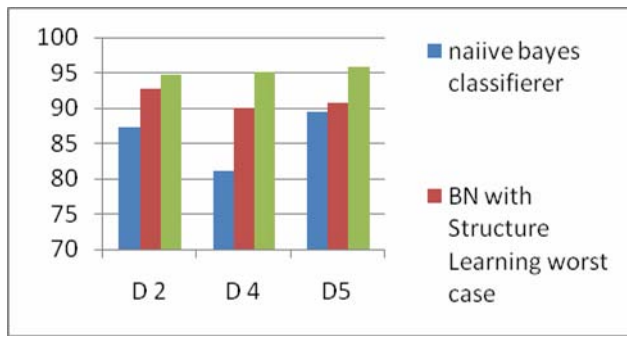


Fig. 3. Comparison of classification accuracy of naive Bayes classifier and Bayesian network using structural learning

5 Conclusions

In this paper, structural learning was proposed for learning the structure of BN classifier. Learning algorithms were implemented and compared for learning CPTs. Proposed algorithms in this paper demonstrate good performances in classification and ranking, and have a relatively low time complexity. As a result, it was shown that from the point of clustering performance, BIC is almost the best measure compared to MDL and AIC scoring measures, but it is the worst algorithm from the point of computation time. K2 is the best learning algorithm according to time complexity but almost the worst one in accuracy compared to other algorithms. The experiments also

show that Bayesian network with structural learning outperforms the Naive Bayes classifier. Based on the results obtained for classification of 5 different datasets, we believe that the BN classifiers can be used more often in many applications. As future works, we want to investigate the use of evolutionary based algorithms for searching the state space in structural learning.

References

1. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, LosAlios (1988)
2. Heckerman, D.: A tutorial on learning with Bayesian networks, Microsoft Technical Report 95-06 (1996)
3. Gyftodimos, E., Flach, P.: Hierarchical Bayesian networks: an approach to classification and learning for structured data, Methods and Applications of Artificial Intelligence. In: Third Hellenic Conference on AI, SETN 2004, Samos, Greece (2004)
4. Friedman, N., Murphy, K., Russell, S.: Learning the Structure of Dynamic Probabilistic Networks. In: Proceedings of the 14th Conference on Uncertainty in AI, pp. 139–147 (1998)
5. Murphy, K.: Dynamic bayesian networks: representation, inference and learning Ph. D. thesis, University of California, Berkeley (2002)
6. Russell, S., Norvig, P.: Artificial intelligence, a modern approach. Prentice Hall, New York (2003)
7. Heckerman, D., Geiger, D., Chickering, D.M.: Learning Bayesian networks: The combination of knowledge and statistical data. Machine Learning 20, 197–243 (1995)
8. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. Mach. Learn. 2, 131–163 (1997)
9. Friedman, N., Goldszmidt, M.: Learning Bayesian networks with local structure. In: Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence, pp. 252–262 (1996)
10. Heckerman, D.: A tutorial on learning with Bayesian networks. In: Jordan, M.I. (ed.) Learning in Graphical Models, pp. 301–354. MIT Press, Cambridge (1999)
11. Lam, W., Bachus, F.: Learning Bayesian Networks. An approach based on the MDL principal. Computational Intelligence 10(3), 269–293 (1994)
12. Cooper, G.: Computational complexity of probabilistic inference using Bayesian belief networks (Research Note). Artificial Intelligence 42, 393–405 (1990)
13. Cooper, G., Herskovitz, E.: A Bayesian Method for Constructing Bayesian Belief Networks from Databases. In: Proceedings of the 7th Conference on Uncertainty in AI, pp. 86–94 (1991)

A Clustering Method Based on Soft Learning of Model (Prototype) and Dissimilarity Metrics

Arash Arami and Babak Nadjar Araabi

Control and Intelligent Processing Center of Excellence
Electrical and Computer engineering Department, University of Tehran, Tehran, Iran
School of Cognitive Sciences, IPM, Tehran, Iran
a.arami@ece.ut.ac.ir, araabi@ut.ac.ir

Abstract. Many clustering methods are designed for especial cluster types or have good performance dealing with particular size and shape of clusters. The main problem in this connection is how to define a similarity (or dissimilarity) criterion to make an algorithm capable of clustering general data, which include clusters of different shape and size. In this paper a new approach to fuzzy clustering is proposed, in which during learning a model for each cluster is estimated. Gradually besides, dissimilarity metric for each cluster is defined, updated and used for the next step. In our approach, instead of associating a single cluster type to each cluster, we assume a set of possible cluster types for each cluster with different grades of possibility. Also proposed method has the capability to deal with partial labeled data. Comparing the experimental results of this method with several important existing algorithms, demonstrates the superior performance of proposed method. The merit of this method is its ability to deal with clusters of different shape and size.

Keywords: Clustering, Cluster prototype, Mass prototype, linear prototype, Shell Prototype, Fuzzy membership function.

1 Introduction

Clustering tries to categorize unlabeled data such that the data in each category have the most similarity to each other and dissimilarity to data in other categories. In the other words, the goal of clustering is to label unlabeled data in a special manner such that the data with the same label are similar to each other and differ from the data with another label. Several applications in different criteria use clustering, for instance machine vision [1], images segmentation [2], image compression [3], and data mining. Generally speaking, clustering is useful wherever finding rational relativity of data is needed.

Since clustering tends to discover data patterns, similarity criteria definition according to different data structures is a very important and hard-working job in its succession. This is more clarified when the data has non-mass type extensions, for example linear type or shell type, in which a wrong similarity/dissimilarity criterion definition leads to an incorrect clustering. We don't have information about data extension types and the clusters' models in general cases; hence a method which can

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 33–40, 2008.

learn, revises its criterion and uses it in next steps during clustering seems to be very useful. In this paper we attempt to propose a method based on learning cluster models and similarity/dissimilarity criterion. Fuzzy clustering, which has been originated by fuzzy sets theory considering uncertainties in finding clusters, is also based on optimization. One of the primary methods to this goal was introduced by Dunn [4], which was based on objective functions defined by Euclidean distances. This method was then extended by Bezdak [5] and [6]. Soft clustering algorithms based on L_1 norm were propounded by Jajuga [7]. Furthermore, Yang has done a thorough survey on fuzzy clustering [8]. There also has been adaptive fuzzy clustering method introduced by Gustafson and Kessel (GK) [9], in which the clusters differ according to a quadratic distance defined in a fuzzy covariance matrix. A detailed study of this algorithm was presented by Krishnapuram and Kim [10], recently a partitional fuzzy clustering method based on adaptive quadratic distance introduced by Carvalho [11]. Also Bouchachia has enhanced the performance of fuzzy clustering by adding a mechanism of partial supervision [12].

In section 2, proposed clustering method is introduced and formulated. The experimental results and comparison with other methods is presented in section 3, finally the conclusions are drawn in last section.

2 Proposed Algorithm

Clustering different type of clusters (Mass, linear, shell type) needs different kind of dissimilarity functions. When there is no prior knowledge about type and shape of clusters, it is important to use a flexible dissimilarity function between them. Some methods, like GK, have a dissimilarity function with limited capability to change it metrics but when the clusters have different sizes, it doesn't work out very well.

In this section an algorithm has been proposed, which has flexible dissimilarity function. This function is a weighted sum of conventional dissimilarity functions of Mass, linear, and shell type fuzzy clustering. During the clustering, the weights of these functions change with respect to how much a cluster belongs to each type. Moreover, a soft switching is applied between each type of dissimilarity function to estimate the proper function of dissimilarity. In addition, the prototype of each cluster is a combination of different prototypes. The algorithm is applicable to both data including or excluding partially labeled data and is presented in the following, but whenever the labeled data are used, the substitutive step for the latter group is performed. The steps of algorithm are presented as follows;

Perform an initial clustering, for example fuzzy C-means (FCM) clustering.

Consider the U_{ij} s corresponding to labeled data equal to 1 and 0 otherwise.

Now find a mass prototype for each cluster according to achieved U_{ij} s.

Compute fuzzy scatter matrix using U_{ij} s, and find a linear prototype for each cluster.

Using μ_i s and data points, which are computed in first step find a radius about μ_i and find a shell prototype for each cluster accordingly.

Form the scatter matrix (SM) for partially labeled data if the ratio of largest eigenvalue to the smallest one, denoted by *Rat*, was smaller than a threshold then consider the initial value of distance weight to mass prototype relatively small.

If there weren't any partially labeled data, then after first step, form the fuzzy SM (FSM) and Perform step 5 for FSM matrix.

As mentioned above, the dissimilarity criterion is a weighted sum of distance to different possible prototypes:

$$Dist\,(x^j,\omega_i) = a_{i1}dist_M\,(x^j,\omega_{imass}) + a_{i2}dist_L(x^j,\omega_{ilinear}) + a_{i3}dist_S(x^j,\omega_{ishell}) \tag{1}$$

in which $a_{i1}+a_{i2}+a_{i3}=1$. In order to satisfy this equation a term need to be added to each a_{ij} s after updating as a post processing.

If *Rat* was smaller than a threshold, the cluster would have mass type. If *Rat* was bigger than another threshold, the cluster would have linear type.

After computing the final distance of each datum to each cluster, which is the linear combination of its distances to cluster prototypes, U_{ij} s will be computed using equation 2.

$$U_{ij} = 1/\sum_{L=1}^{c}\left(Dist\,(x^j,\omega_i)/Dist\,(x^j,\omega_L)\right)^{\frac{1}{m-1}} \quad ; \quad m>1 \tag{2}$$

Then a_1, a_2, a_3, which are multipliers of dissimilarity metric, are updated according to the following formula.

$$a_{i1}(t+1) = a_{i1}(t) + \eta \times a_{i1}(t)U_{ij}(t) \times \left(MLSd - dist_M\,(x^j,\omega_{imass})\right)/Sd \tag{3}$$

In which;

$$MLSd = \left(dist_L^{\,*}(x^j,\omega_{ilinear}) + dist_S(x^j,\omega_{ishell})\right)/2 \tag{4}$$

$$Sd = dist_M\,(x^j,\omega_{imass}) + dist_L^{\,*}(x^j,\omega_{ilinear}) + dist_S(x^j,\omega_{ishell}) \tag{5}$$

$$\eta = \frac{1}{N} \tag{6}$$

The formula (3) is the updating rule for a_{i1} weight and the ones for a_{i2} and a_{i3} can be achieved easily substituting the numerator terms. $dist_L^{\,*}$ is described in next part.

Different prototypes are updated then according to achieved U_{ij} s. This updating is similar to FCM method for mass type, Fuzzy C-Linear (FCL) method for linear type, and Fuzzy C-Spherical (FCS) method for shell type. Steps 6 to 9 are performed iteratively to achieve the appropriate solution.

The grade of possibility of each cluster type or in the other hand, degree of membership of each cluster to mass, linear or shell type model is obtained through the following formulas:

$$M_{mass}(i) = a_{i1}/(a_{i1} + a_{i2} + a_{i3}) = a_{i1}$$
$$M_{linear}(i) = a_{i2}/(a_{i1} + a_{i2} + a_{i3}) = a_{i2} \quad (7)$$
$$M_{shell}(i) = a_{i3}/(a_{i1} + a_{i2} + a_{i3}) = a_{i3}$$

2.1 Distance Correction Coefficient for Different Prototypes

The distance to a linear prototype, as shown in (Fig. 1), is less than the distance to a mass prototype even if the cluster has the mass form. This problem can be solved, using a coefficient in distance formula.

Data on the surface, which meets the sphere centre and is perpendicular to linear prototype direction, have equal Euclidean distance from the both mass and linear prototypes; but, other points inside the sphere have less distance to supposed linear prototype. We have:

$$dist_M^{\,2} = dist_L^{\,2} + r'^2; \quad 0 \le r' \le R \quad (8)$$

Where, R is the sphere radius, $dist_M$ is distance from mass prototype, and $dist_L$ is distance from linear prototype.

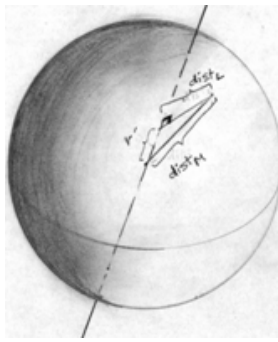


Fig. 1. Mass-type cluster model with supposed prototypes

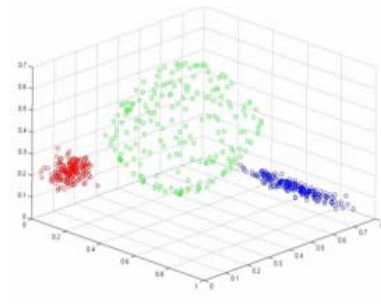


Fig. 2. 3D Data in the normalized form

Now, for avoiding error in specialize contributions to different prototypes, K corrects the distance in the following manner:

$$dist_L^{*} = \sqrt{dist_L^{\,2} + K} \quad (9)$$

For instance, K for a uniform data distribution in 3D feature space is calculated as below in formula (10). More discussion about correction coefficient is presented in Appendix. It is noticeable to state that R must not be the maximum distance of data to the sphere centre, which can effect the robustness of the method; It should define the mass logical radius, for example for Gaussian distribution σ or 2σ where σ is the variance of data in mass cluster.

$$K = (3/4\pi R^3) \times 2\int_0^R \pi r^2 (R^2 - r^2) dr = R^2/5 \quad (10)$$

3 Experimental Results

The dataset used for testing this method has different clusters of mass, linear, and shell type. Fig. 2, shows the normalized data. This dataset has three groups of data including two groups with Gaussian distribution (blue linear-type cluster and red mass-type cluster) and a group with uniform distribution (green shell-type cluster).

The proposed algorithm is applied to this dataset and is compared with several clustering algorithms. The results are illustrated in the following figures (Fig. 3, Fig. 4). For quantitative comparisons between mentioned methods, some clustering validity indices are utilized:

Partition Coefficient, which is described below:

$$PC = \sum_{i=1}^{C} \sum_{j=1}^{N} U_{ij}^{\ 2} / |X| \quad ; \quad |X| = N \tag{11}$$

It is obvious that (1/C)<PC<1 and bigger PC is equivalent to crisper clustering.

Partition Entropy; this is described below:

$$PE = -\sum_{i=1}^{C} \sum_{j=1}^{N} U_{ij} \ Ln(U_{ij}) / |X| \quad ; \quad |X| = N \Rightarrow 0 \le PE \le \ln(c) \tag{12}$$

The smaller is PE, the crisper is the clustering performance.

Separation D, which is described below:

$$D = \min_{\substack{i=1 \\ I \neq J}}^{C} \min_{j=1}^{C} dist_{\min}(\omega_i, \omega_j) / \max_{l=1}^{C} dist_{\max}(\omega_l, \omega_l) \tag{13}$$

The greater is D, the more valid is the clustering.

Separation S, which is described below:

$$S = \min_{\substack{i=1 \\ I \neq J}}^{C} \min_{j=1}^{C} dist_{\min}(\omega_i, \omega_j) / \max_{l=1}^{C} dist_{\max}(\omega_l, \omega_l) \tag{14}$$

The smaller is S, the more valid is the clustering.

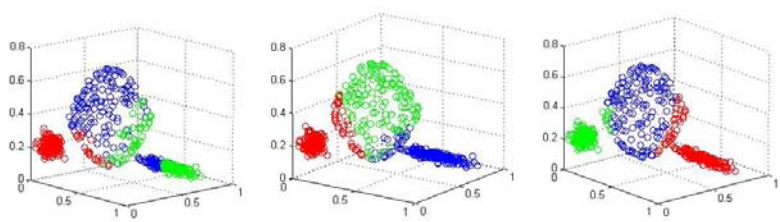


Fig. 3. Data clustered using fuzzy c-spherical clustering (on the left), fuzzy c-linear clustering (on the middle) and fuzzy C-means (on the right)

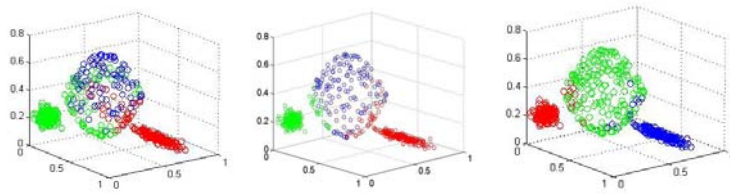


Fig. 4. Gustafson-Kessel clustering (on the left), proposed algorithm without partially labeled data (on the middle) and proposed algorithm with partially labeled data observations (on the right)

Table 1. Quantitative comparison between different clustering methods

Validity index	Cluster validity index	Fuzzy, mass proto.	Fuzzy C-Linear, linear proto.	Fuzzy C-Spherical, shell proto.	Fuzzy C-Means	GK	Proposed Algorithm, no P.L.O.	Proposed Algorithm, P.L.O.
Expected Value	PC	0.692	0.791	0.890	0.781	0.817	0.669	0.684
	PE	0.560	0.385	0.203	0.402	0.345	0.357	0.343
	D	0.699	0.385	0.611	1.278	0.087	0.787	0.770
	S	127.3	906.3	137.7	89.07	858.7	67.129	62.442
Standard Deviation	PC	0.001	5.7E-5	3E-4	6E-4	0.006	0.0155	0.074
	PE	0.002	4.0E-4	0.002	8E-4	0.012	0.023	0.063
	D	0.198	0.296	0.012	0.012	0.015	0.136	0.185
	S	38.93	1687	3.945	1.332	121.9	18.776	29.182

Each of the mentioned algorithms is executed 30 times on the dataset and the results are provided in table 1. As can be deduced from table 1, the proposed method, especially in partially labeled observations attendance, and also the FCM method have succeeded more than the others. The proposed method also has ended to good separation S validity, while FCM has lead into good separation D validity.

Both algorithms do not much differ in PE and PC. The later indices, PE and PC are not individually support the quality of clustering and just describe the crispness of clustering. Since the indices in standard deviation without P.L.Os decreases, it can be deduced that initializing a_1, a_2, a_3, should be improved. It is noticeable that this method has some computational complexities compared with same fuzzy clustering methods family and its performance decreases dealing with clusters, which are far from each other. Also it is notable to state that the GK powerful method performs weakly due to presence of different cluster sizes.

4 Conclusion

Many clustering methods are designed for especial cluster types or have good performance dealing with particular size and shape of clusters. This paper presented a

new approach to fuzzy clustering, in which during learning a model for each cluster estimated. Dissimilarity metric for each cluster is defined, updated and used for the next step. Its strength in dealing with clusters of different type and size is the most important advantage of this method. Also our approach has the capability to deal with partial labeled data as well as fully unlabeled data. This method is implemented on two families of data, first in presence of partially labeled data (10% of data are labeled) and second, with fully unlabeled data. Comparing the experimental results of this method with several important existing algorithms verified its succession both in achieving good value of clustering indexes and to estimating each cluster shape. Comparing with different pattern recognition methods which convert the feature space into a space with more dimensions, the proposed method has the capability of computing a fuzzy membership value to different shapes for each cluster in its basic feature space. This capability can make the method useful in shape recognition tasks.

References

1. Frigui, H., Krishnapuram, R.: A Robust Competitive Clustering Algorithm with Applications in Computer Vision. IEEE Trans. Pattern Analysis and Machine Intelligence 21(5), 450–465 (1999)
2. Rezaee, M.R., Van Der Zwet, P.M.J., Lelieveldt, B.P.F., Van Der Geest, R.J., Reiber, J.H.C.: A Multiresolution Image Segmentation Technique Based on Pyramidal Segmentation and Fuzzy Clustering. IEEE Trans. Image Proc. 9(7), 1238–1248 (2000)
3. Kaarna, A., Zemcik, P., Kalviainen, H., Parkkinen, J.: Compression of Multispectral Remote Sensing Images using Clustering and Spectral Reduction. IEEE Trans. Geoscience and Remote Sensing 38(2 II), 1073–1082 (2000)
4. Dunn, J.C.: A Fuzzy Relative to the ISODATA Process and Its Use in Detecting Compact, Well-Separated Clusters. J. Cybernet. 3, 32–57 (1974)
5. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981)
6. Hathaway, R.J., Bezdek, J.C.: NERF c-means: Non Euclidean Relational Fuzzy Clustering Algorithms. Pattern Recognition 2, 429–437 (1994)
7. Groenen, P.J.F., Jajuga, K.: Fuzzy Clustering with Squared Minkovsky Distances. Fuzzy Sets and Systems 120, 227–237 (2001)
8. Yang, M.S.: A Survey of Fuzzy Clustering. Math. Computing. Modeling. 18(11), 1–16 (1993)
9. Gustafson, D.E., Kessel, W.: Fuzzy Clustering with Fuzzy Covariance Matrix. In: Proc. IEEE conf. Decision Contr., San Diego, CA, pp. 761–766 (1979)
10. Krishnapuram, R., Kim, J.: A Note on the Gustafson-Kessel and Adaptive Fuzzy Clustering Algorithms. IEEE Trans. Fuzzy Systems 7(4), 453–461 (1999)
11. Carvalho, F.A.T., Tenório, C.P., Cavalcanti Junior, N.L.: Partitional Fuzzy Clustering Methods Based on Adaptive Quadratic Distances. Fuzzy Sets and Systems 157, 2833–2857 (2006)
12. Bouchachia, A., Pedrycz, W.: Enhancement of Fuzzy Clustering by Mechanism of Partial Supervision. Fuzzy Sets and Systems 157, 1733–1759 (2006)

Appendix: More Discussion about Distance Correction Coefficient

Assuming that each mass of data in n-dimension feature space consists of infinite number of n-1 dimensional surfaces of data. Attending to Fig. 5, without loosing any generality and for ease of description, assume a 3D space and a spherical mass data with maximum distance from its prototype equal to R_{max}. With respect to distribution of data on this mass a logical radius of sphere can be chosen which denoted by *R*. By slicing the mass orthogonally to the assumed linear prototype direction, infinite number of surfaces are achieved which drawn by dotted lines. Using equations (8) and (9) for each surface, the correction constant for each datum on each surface easily calculated as below;

$$K_{ij} = R_{ij}^{\ 2} - r_i^{\ 2} \tag{15}$$

r_i is the *i*th surface radius and $R_{i,j}$ is the *j*th datum in *i*th surface distance from mass prototype. With assuming that the number of data in each surface is a function of data distribution and area of surface the constant for all data in each surface calculated as below;

$$K_i = \mathrm{F}(\mathrm{distribution\text{-}of\text{-}data\ in\ mass, area\ of\ surface\ i}) \times (\mathrm{R}^2 - \mathrm{r}_\mathrm{i}^2) \tag{16}$$

If there is no information of distribution the effect of this parameter could be neglected or distribution assumed as Gaussian. And the constant for total mass is computed as:

$$K = (1/(\mathrm{mass\ \ volume})) \times \sum_{all\ i | r_i < R} K_i \tag{17}$$

For example with assuming the uniform distribution of data and in 3D feature space, *K* calculated as below;

$$K = (3/4\pi R^3) \times 2\int_0^R \pi r^2 (R^2 - r^2)dr = R^2/5 \tag{18}$$

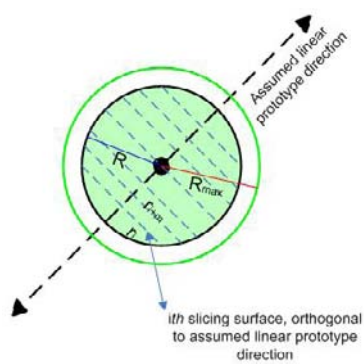


Fig. 5. 2D cut of 3D data and infinite slicing surfaces

An Approximation Algorithm for the *k*-Level Uncapacitated Facility Location Problem with Penalties

Mohsen Asadi[1], Ali Niknafs[1], and Mohammad Ghodsi[1,2]

[1] Computer Engineering Department, Sharif University of Technology, Tehran, Iran
[2] School of Computer Science, Institute for Studies in Theoretical Physics and Mathematics, Tehran, Iran
{mo_asadi,niknafs}@ce.sharif.edu, ghodsi@sharif.edu

Abstract. The *k*-level Uncapacitated Facility Location (UFL) problem is a generalization of the UFL and the *k*-median problems. A significant shortcoming of the classical UFL problem is that often a few very distant customers, known as outliers, can leave an undesirable effect on the final solution. This deficiency is considered in a new variant called UFL with outliers, in which, in contrast to the other problems that need all of the customers to be serviced, there is no need to service the entire set of customers. UFL with Penalties (UFLWP) is a variant of the UFL with outliers problem in which we will decide on whether to provide service for each customer and pay the connection cost, or to reject it and pay the penalty. In this paper we will propose a new 4-approximation algorithm for the UFLWP which is the first algorithm for this kind of problem.

Keywords: Facility Location, Uncapacitated Facility Location, *k*-level Facility Location, Facility Location with Outliers.

1 Introduction

Facility location problem is a fundamental problem in theoretical science. In the classical single-level Uncapacitated Facility Location (UFL) problem, we are given two sets: one set for facilities (F), and another one for customers (C). There is a specified connection cost $c_{ij} \geq 0$ between every pair $i,j \in F \cup C$. Opening a facility $i \in F$ causes a fixed nonnegative open cost f_i. The goal of this problem is to locate facilities and assign each customer to one of the opened facilities, such that the total cost of opening facilities and of servicing the customers would be minimized. That is, in a formal manner, we should identify a subset of facilities $S \subseteq F$ and to serve the customers in C by the facilities in S, so as to minimize the total cost function:

$$Cost(S) = \sum_{i \in S} f_i + \sum_{\substack{i \in S \\ j \in C}} c_{ij}$$

In this paper, we adopt notions of two variants of uncapacitated facility location problem and mix them to form a new problem, known as *k-level Uncapacitated Facility Location with Outliers* problem. The problem of the metric *k*-level UFL problem

[1] This author's work was in part supported by a grant from IPM (N. CS2386-2-01.)

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 41–49, 2008.

is a generalization of the UFL and the k-median problems. In this problem, each customer must be serviced by a sequence of k different kinds of facilities located in k levels of hierarchy. This well-known problem has various applications which can be found in [1].

Another variant of the facility location problem is called UFL with outliers. In this kind of problem, in contrast to the forementioned other problems in which all of the customers must be serviced, there is no need to service the entire set of customers. This is due to the fact that often a few very distant customers, called *outliers*, can leave an undesirable effect on the result solution. This deficiency is considered in a new variant called UFL with outliers, in which very distant outlier may be ignored due to economical issues [2]. Two variants of the UFL with outliers problem have been proposed in the literature:

- *Uncapacitated Facility Location with Penalties (UFLWP)*: in this problem a penalty r_j is assigned to each customer$j$. For each customer we may decide to either provide service, and pay the connection cost to its nearest facility or to ignore it and pay the penalty. Setting the penalties to ∞ gives the standard formulation.
- *Robust Facility Location (RFL)*: in this problem we are given a parameter η. The problem is to locate facilities so as to minimize the service cost to any subset of facilities of size at least η. In case that $\eta = n$ the problem is equivalent to the standard formulation.

See [3] for a comprehensive review on other variants of the facility location problem with outliers. A number of efficient approaches have been proposed in recent years, which can be roughly classified into several categories: greedy approach [4], LP rounding techniques [5], local search heuristics [6], primal-dual method [5], game theory [7], and randomization technique [8]. In some sense, techniques from different categories complement each other, and could be combined to achieve improved approximation algorithms [4]. The first approximation algorithm for metric k-level UFL problem proposed in [9] uses the primal-dual scheme in linear programming which has an approximation factor of 6. Another algorithm using game theory achieves the same approximation factor [7]. The approximation factor is improved to 4 using the greedy approach [10]. Aardal et. al. [8] proposes a 3-approximation algorithm using linear programming relaxation.

For UFL with outliers problem three constant approximation algorithms have been proposed [11, 5, 9]. A 3-approximation algorithm by means of primal-dual scheme has been suggested for both kinds of UFL with outliers problem [11]. For the UFLWP problem, an algorithm using the LP rounding technique has been proposed in [5] with $2 + 2/e$ approximation factor, where e is the natural logarithmic base, whereas a 2-approximation algorithm is proposed in [4].

In this paper we examine the metric k-level uncapacitated facility location problem with outliers. This problem is considered as an extension to the classical k-level UFL problem in which the outliers are ignored to achieve an improved level of service to majority of customers. We adopt the UFLWP variant of the single-level UFL with

outliers problem [5], and propose new algorithms for the case with k-levels of facilities by applying the LP techniques. To our knowledge, there is no algorithm proposed for k-level UFL with penalties.

The paper is organized as follows. Next section includes a brief description of the k-level UFL with penalties. Section 3 presents our proposed 4-approximation algorithm for the UFLWP problem.

2 Problem Description

In this section, first we present a formal and precise definition of the k-level UFL problem. Then we explain the required changes need to be performed to obtain the k-level version of UFLWP problem. Finally we formulate the UFLWP by taking the advantage of linear programming technique.

2.1 k-Level UFL

Let C be the set of customers. Each customer $j \in C$ must be assigned to precisely one facility at each of the k levels. Let F^l be the set of locations where facilities on el $l, 1 \leq l \leq k$, may be located and assume that the sets $F^l$are pairwise disjoint. $F = \cup_{l=1}^{k} F^l$ is considered as the set of all such locations. The cost of setting up a facility at location i is f_i, which $f_i > 0$ for each $i \in F$. The cost of connection between points $i, j \in F \cup C$ is equal to c_{ij}, which $c_{ij} > 0$. Throughout this paper, we make the metric assumption of the k-level UFL problem, i.e. for each $i, j, k \in F \cup C$ we have $c_{ik} \leq c_{ij} + c_{jk}$, which satisfies the triangle inequality. We shall use p to denote a sequence of facilities $i_l \in F^l, l = 1, \dots, k$, and shall refer to p as a *path* of facilities. The set of all possible paths is denoted by P. each customer must be assigned to precisely one path $p \in P$. The total connection cost incurred by assigning customer j to path $p = (i_1, i_2, \dots, i_k)$ is equal to $c_{jp} = c_{i_1,i_2} + c_{i_2,i_3} + \cdots + c_{i_{k-1},i_k} + c_{i_k,j}$.

The goal of this problem is to assign each customer to a sequence of k facilities, one at each level, such that satisfying the service demand of each customer and minimize the total cost of opening facilities and connection costs.

2.2 k-Level UFLWP

This problem is derived from the classical k-level UFL problem. Each customer $j \in C$ will be either serviced or rejected completely. If the customer is planned to be serviced then it should be assigned to a sequence of facilities one in each level. An r_j parameter is assigned to each customer $j \in C$ indicating the penalty of rejecting a customer. The goal of this problem is to find:

1. A subset $Q \subseteq C$ of customers whose demands should be rejected,
2. A subset $S \subseteq F$ as the locations for opening facilities, such that $S = \cup_{l=1}^{k} S^l$, and
3. Remaining customers are assigned to a sequence of k facilities, one at each level, so that each customer will be serviced.

The following is a formulation for the k-level UFLWP by means of integer programming (denoted as IP_1). Let suppose x_{jp} equals one, if customer j is assigned to path p, otherwise it equals zero. In addition, let y_{i_l} equals 1 if the facility $i_l$on level l is open. Furthermore z_j equals 1 if the customer is rejected and 0 if the customer is serviced. Equations (1)-(6) represent the integer program of this problem.

$$minimize$$

$$\boldsymbol{Z_{IP_1} = \sum_{l=1}^{k} \sum_{i_l \in F^l} f_{i_l} y_{i_l} + \sum_{p \in P} \sum_{j \in C} c_{jp} x_{jp} + \sum_{j \in C} r_j z_j} \quad (1)$$

$$\boldsymbol{subject\ to}$$

$$\boldsymbol{z_j + \sum_{p \in P} x_{jp} \geq 1 \quad for\ each\ j \in C} \quad (2)$$

$$\boldsymbol{\sum_{P:p \ni i_l} x_{jp} - y_{i_l} \leq 0}$$
$$\boldsymbol{for\ each\ j \in C\ and\ i_l \in F^l, l = 1, \dots, k} \quad (3)$$

$$\boldsymbol{x_{jp} \in \{0,1\}}$$
$$\boldsymbol{for\ each\ j \in C\ and\ i_l \in F^l, l = 1, \dots, k} \quad (4)$$

$$\boldsymbol{y_{i_l} \in \{0,1\} \quad for\ each\ i_l \in F^l, l = 1, \dots, k} \quad (5)$$

$$\boldsymbol{z_j \in \{0,1\} \quad for\ each\ j \in C} \quad (6)$$

3 Algorithm

In this section, we present a constant factor approximation algorithm for the k-level UFLWP problem, by adopting the *LP rounding technique*. The linear relaxation version of $IP_1$is obtained by relaxing the constraints (4), (5), and (6). That is these constraints are replaced with inequalities $x_{jp} \geq 0, y_{i_l} \geq 0,$ and $z_i \geq 0$ respectively. Our solution to the UFLWP problem, first solves the linear relaxation program for each $j \in C$. Let LP_1 denote this relaxation program, which will be solved by means of existing LP solving techniques. The obtained solution would be fractional; however, we need to round the solution to achieve a near optimum integral solution suitable for the actual integer program. Formulation of the LP_1 relaxation program is as follows:

$$\boldsymbol{minimize}$$
$$\boldsymbol{Z_{LP_1} = \sum_{l=1}^{k} \sum_{i_l \in F^l} f_{i_l} y_{i_l} + \sum_{p \in P} \sum_{j \in C} c_{jp} x_{jp} + \sum_{j \in C} r_j z_j} \quad (7)$$

$$\boldsymbol{subject\ to}$$
$$\boldsymbol{z_j + \sum_{p \in P} x_{jp} \geq 1\ \ for\ each\ j \in C} \quad (8)$$

$$\sum_{P:p\ni i_l} x_{jp} - y_{i_l} \le 0$$
$$\textbf{for each } j \in C \textbf{ and } i_l \in F^l, l = 1, \dots, k \quad (9)$$

$$x_{jp} \ge 0 \quad \textbf{for each } j \in C \textbf{ and } p \in P \quad (10)$$

$$y_{i_l} \ge 0 \quad \textbf{for each } i_l \in F^l, l = 1, \dots, k \quad (11)$$

$$z_j \ge 0 \quad \textbf{for each } j \in C \quad (12)$$

Note that in the above program, due to the constraints (8) and (10), there is no need to include the upper bound $x_{jp} \le 1$.

Let (x', y', z')be the solution of LP_1, and w'be the optimal objective value of it. Likewise, suppose that OPT be the optimal objective value of IP_1. Obviously, $OPT \ge w'$.

After solving the linear program and obtaining the above values, the customers whom should be rejected are then identified. This is done by comparing the value of $z'_j$of each customer $j \in C$ with a threshold. Let $\delta \in (0,1)$ be this threshold. Moreover, let $Q_s$be the set of rejected customers, i.e. $Q_s = \{j \in C \mid z'_j \ge \delta\}$. If $Q_s = C$, then every customer must be rejected, so the algorithm is terminated. Otherwise, the remaining customers need to be serviced via opening new facilities.

In order to define an upper bound for the obtained solution, we introduce auxiliary variables $\bar{x}$ and $\bar{y}$. It is shown in the Lemma 1 how they can be used to analyze the approximation factor of the algorithm. More specifically, two auxiliary variables are defined as follows:

$$\bar{x}_{jp} = \begin{cases} 0 & \textbf{if } j \in Q_s \\ \min\left(\frac{1}{1-\delta} x'_{jp}, 1\right) & \textbf{if } j \notin Q_s \end{cases} \quad (13)$$

$$\bar{y}_{i_l} = \min\left(\frac{1}{1-\delta} y'_{i_l}, 1\right)$$
$$\textbf{for each } i_l \in F^l, l = 0, \dots, k \quad (14)$$

To find a solution with the minimum cost of opening facilities and of assigning customers which are selected to be serviced, i.e. those who are not placed in Q_s, we introduce the following integer program (denoted as IP_2) for the new UFL problem with F as the set of facilities and $\mathcal{C} = \{j \in C \mid j \notin Q_s\}$ as the set of customers. After relaxing the constraints,

$$x_{jp} \in \{0,1\} \quad for\ each\ j \in \mathcal{C}\ and\ i_l \in F^l, l = 1, \dots, k$$
$$y_{i_l} \in \{0,1\} \quad for\ each\ i_l \in F^l, l = 1, \dots, k$$

the linear relaxation of IP_2 is obtained (denoted as LP_2), which is presented as follows:

$$\boldsymbol{minimize}$$

$$\boldsymbol{Z_{LP_2} = min \sum_{l=1}^{k} \sum_{i_l \in F^l} f_{i_l} y_{i_l} + \sum_{p \in P} \sum_{j \in \mathcal{C}} c_{jp} x_{jp}} \tag{15}$$

$$\boldsymbol{subject\ to}$$

$$\boldsymbol{\sum_{p \in P} x_{jp} \geq 1 \quad for\ each\ j \in \mathcal{C}} \tag{16}$$

$$\boldsymbol{\sum_{P: p \ni i_l} x_{jp} - y_{i_l} \leq 0 \quad for\ each\ j \in \mathcal{C}\ and\ i_l \in F^l, l = 1, \ldots, k} \tag{17}$$

$$\boldsymbol{x_{jp} \geq 0 \quad for\ each\ j \in \mathcal{C}\ and\ p \in P} \tag{18}$$

$$\boldsymbol{y_{i_l} \geq 0 \quad for\ each\ i_l \in F^l, l = 1, \ldots, k} \tag{19}$$

Let w^* be the optimal objective value of LP_2 and let A be an approximation algorithm to solve IP_2 for the k-level UFL problem (such as [8]). Approximation factor of A would be $= \frac{A^*}{w^*}$, where A^* is the objective value of the integral solution returned by algorithm A. Since, $OPT_A > w^*$, then the approximation factor of algorithm A, i.e. $\frac{A^*}{OPT_A}$, will be no larger than λ. Therefore, the total cost of opening facilities and connections between customers and opened facilities, returned by algorithm A is bounded by λw^*.

By applying algorithm A to IP_2, customers belonging to $\mathcal{C}$ are assigned to facilities in F. Afterwards, the obtained solution is combined with the set of rejected customers (those belonging to Q_s) to form a solution to IP_1. The total cost of the latest obtained solution is no more than $\lambda w^* + \sum_{j \in Q_s} r_j$.

In the following describe the algorithm and prove its approximation factor.

Algorithm 1 (Metric k-level UFLWP – factor 4)

1. Solve the linear program LP_1 for facility set F and customer set C.
2. Find the set of customers to be rejected (Q_s), and reject them, if all customers are rejected then terminate, Define $\mathcal{C} = C \backslash Q_s$.
3. Apply approximation algorithm A to the problem with F as facility set and $\mathcal{C}$ as customer set. The algorithm opens facilities in F and assigns customers belonging to $\mathcal{C}$ according to the solution returned by A.

To determine the approximation factor we need to find the relation between the solution obtained for IP_1 and IP_2. This relationship can be identified by means of the next two lemmas.

Lemma 1. Let $(\bar{x}, \bar{y})$ be a feasible solution for LP_2 and,

$$\boldsymbol{w^* \leq \frac{1}{1-\delta} \sum_{l=1}^{k} \sum_{i_l \in F^l} f_{i_l} y'_{i_l} + \frac{1}{1-\delta} \sum_{p \in P} \sum_{j \in C} c_{jp} x'_{jp}} \tag{20}$$

Proof. First, we need to show that $(\bar{x}, \bar{y})$ is a feasible solution for LP_2. We know that $\bar{x}_i$ and $\bar{y}_i$ always have positive values. So, we only need to show that constraints (16) and (17) are correct.

Suppose an arbitrary customer $j \notin Q_s$, we have:

$$\sum_{p\in P} \bar{x}_{jp} = \sum_{p\in P} \min\left(\frac{1}{1-\delta}\, x'_{jp}, 1\right) \tag{21}$$

If there exists a path $p \in P$, such that $\frac{1}{1-\delta}\, \bar{x}_{jp} \geq 1$, the constraint (16) would be obviously correct. Otherwise,

$$\begin{aligned}\sum_{p\in P} \min\left(\frac{1}{1-\delta}\, x'_{jp}, 1\right) &= \sum_{p\in P} \frac{1}{1-\delta}\, x'_{jp} \\ &\geq \frac{1}{1-\delta}\left(1 - z'_j\right)\end{aligned} \tag{22}$$

The above inequality is derived from constraint (2) and the fact that (x', y', z') is feasible solution for LP_1. Since we have $j \notin Q_s$, then $z'_j \leq \delta$. Therefore,

$$\begin{aligned}\sum_{p\in P} \min\left(\frac{1}{1-\delta}\, x'_{jp}, 1\right) &\geq \frac{1}{1-\delta}\left(1 - z'_j\right) \\ &\geq 1\end{aligned} \tag{23}$$

and so the constraint (16) would be correct.

To prove the correctness of the constraint (18), we have $x'_{jp} \leq y'_{i_l}$ due to the constraint (3), so,

$$\bar{x}_{jp} = \min\left(\frac{1}{1-\delta} x'_{jp}, 1\right) \leq \min\left(\frac{1}{1-\delta} y'_{i_l}, 1\right) = \bar{y}_{i_l} \tag{24}$$

Now we have to bound the value of w^*. Since, $(\bar{x}, \bar{y})$ is a feasible solution for LP_2, and $w^*$is the optimal solution for LP_2, then w^* would be the lower bound of the objective value of $(\bar{x}, \bar{y})$. Thus,

$$w^* \leq \sum_{l=1}^{k} \sum_{i_l \in F^l} f_{i_l} \bar{y}_{il} + \sum_{p\in P} \sum_{j\in C} c_{jp} \bar{x}_{jp} \tag{25}$$

By the definitions of $\bar{x}$ and $\bar{y}$, we have:

$$\begin{aligned} w^* &\leq \sum_{l=1}^{k} \sum_{i_l \in F^l} \frac{1}{1-\delta} f_{i_l} y'_{i_l} + \sum_{p\in P} \sum_{j\in C} \frac{1}{1-\delta} c_{jp} x'_{jp} \\ &\leq \sum_{l=1}^{k} \sum_{i_l \in F^l} \frac{1}{1-\delta} f_{i_l} y'_{i_l} + \sum_{p\in P} \sum_{j\in C} \frac{1}{1-\delta} c_{jp} x'_{jp} \end{aligned} \tag{26}$$

Lemma 2. Suppose that A is the approximation algorithm with factor λ. Then the above algorithm for the metric k-level UFLWP problem is a $(1+\lambda)$-approximation algorithm.

Proof. As we stated before the cost of algorithm is:

$$\begin{aligned} &\lambda w^* + \sum_{j\in Q_s} r_j \\ &\leq \frac{\lambda}{1-\delta}\left(\sum_{l=1}^{k} \sum_{i_l \in F^l} f_{i_l} y'_{i_l} + \sum_{p\in P} \sum_{j\in C} c_{jp} x'_{jp}\right) + \sum_{j\in Q_s} r_j \end{aligned}$$

$$\leq \frac{\lambda}{1-\delta}\left(\sum_{l=1}^{k}\sum_{i_l \in F^l} \boldsymbol{f}_{i_l} \boldsymbol{y}'_{i_l} + \sum_{p \in P}\sum_{j \in C} \boldsymbol{c}_{jp} \boldsymbol{x}'_{jp}\right) + \sum_{j \in Q_s} \frac{1}{\delta} \boldsymbol{r}_j \boldsymbol{z}'_j$$
$$\leq \max\left(\frac{\lambda}{1-\delta}, \frac{1}{\delta}\right) \boldsymbol{w}'$$
$$\leq \max\left(\frac{\lambda}{1-\delta}, \frac{1}{\delta}\right) \boldsymbol{OPT}. \quad (27)$$

If we take $\delta = \frac{1}{1+\lambda}$,the approximation algorithm would be $1 + \lambda$.

Theorem 1. There is a polynomial time algorithm with approximation factor of 4 for the k-level UFLWP.

Proof. Since the algorithm presented in [1], which is supposed the algorithm A with approximation factor 3, so we can use it in step 3 of our algorithm. Therefore, due to lemma 2, the approximation factor of our algorithm is 4 for solving the k-level UFLWP.

4 Discussion

In this paper, we adopt notions of two variants of the uncapacitated facility location problem and mix them to form a new problem, known as *k*-level uncapacitated facility location with outlier. Afterwards, we represent the algorithm for the k-level UFLWP variant of it using LP technique. The approximation factor of the algorithm is 4. To our knowledge this is the only existing algorithm for this problem.

References

1. Sahin, G., Sural, H.: A review of hierarchical facility location models. J. Computers & Operations Research 34 (8), 2310--2331 (2007)
2. Arya, V., Garg, N., Khandekar, R., Munagala, K., Pandit, V.: Local search heuristic for k-median and facility location problems. In: Proceedings of the Thirty-Third Annual ACM Symposium on theory of Computing (STOC '01), Hersonissos, Greece, pp.21--29. ACM Press, New York (2001)
3. Shmoys, D.: Approximation Algorithms For Facility Location Problems. In: Jansen, K., Khuller, S. (eds.) APPROX 2000. LNCS, vol. 1913, pp. 27–32. Springer, Heidelberg (2000)
4. Jain, K., Mahdian, M., Markakis, E., Saberi, A., Vazirani, V.: Greedy facility location algorithms analyzed using dual fitting with factor-revealing LP. J. ACM 50 (6), 795--824 (2003)
5. Xu, G., Xu, J.: An LP rounding algorithm for approximating uncapacitated facility location problem with penalties. J. Information Processing Letters (94), pp.119--123 (2005)
6. Korupolu, M. R., Plaxton, C. G., Rajaraman, R.: Analysis of a local search heuristic for facility location problems. In: Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1--10 (1998)
7. Xu, D., Du, D.: The k-level facility location game. J. Operations Research Letters 34 (4), 421--426 (2006)

8. Aardal, K., Chudak, F. A., Shmoys, D. B.: A 3-approximation algorithm for the k-level uncapacitated facility location problem. J. Information Processing Letters 72 (5-6), 161--167 (1999)
9. Jain, K., Vazirani, V., Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation. J. ACM (48), 274--296 (2001)
10. Zhang, P.: A new approximation algorithm for the k-facility Location Problem. In: Cai, J.-Y., Cooper, S.B., Li, A. (eds.) TAMC 2006. LNCS, vol. 3959, pp. 217–230. Springer, Heidelberg (2006)
11. Charikar, M., Khuller, S., Mount, D., Narasimhan, G.: Algorithms for facility location problems with outliers. In: Proceedings of Symposium on Discrete Algorithms, pp. 642--651 (2001)

Optimizing Fixpoint Evaluation of Logic Programs with Uncertainty

Nematollaah Shiri and Zhi Hong Zheng

Concordia University, Dept. of Computer Science and Software Engineering
1455 De Maisonneuve Blvd. West, Montreal, Quebec, H3G 1M8, Canada
{Shiri,Zh_Zheng}@cse.concordia.ca

Abstract. We investigate efficient techniques for bottom-up, fixpoint evaluation of logic programs and deductive databases with uncertainty over the certainty domain [0,1], often assumed to be a complete lattice ordered by ≤ with min and max as the meet and join operators, respectively. Standard evaluation methods are inadequate in our context in particular when multiset is used as the semantics structure and when programs use aggregate functions other than the lattice join. We propose a semi-naïve method which adopts and extends relation partitioning and backtracking techniques used in standard case. We developed a running prototype of our method, called SNPB, and studied its performance. Our experimental results indicated a speed-up gain, over the semi-naïve method, ranging from 1.25 to 203, depending on the structures and sizes of the input data set and the programs.

Keywords: Uncertainty, Deductive Databases, Multisets, Semantics, Performance.

1 Introduction

Uncertainty management has been among the challenges issues in database and artificial intelligence research for along time. As identified in the Lowell report [1], at the very least, probabilistic reasoning and other techniques for managing uncertainty must become first-class citizen of the DBMS. Standard logic programming [12] and deductive databases [3], [15] with declarative and modularity advantages and with their powerful query processing techniques have been recognized as desired for modeling and reasoning with uncertainty. This resulted in a number of proposals for frameworks with uncertainty, including e.g., [4], [5], [6], [8], [9], 10], [13], [19]. Typically these proposals offer a formalism in which deduction can be combined with some form of uncertainty, including, e.g. certainty values, fuzzy, probabilities, possibilities, etc. As in standard logic programs, these frameworks enjoy a declarative semantics. On the operational side, this is supported by a fixpoint semantics and a corresponding sound and complete (or weakly complete) proof procedure.

These frameworks either use *sets* or *multisets* as the basis for their semantics structures. In the latter case, it has been shown that standard query processing tools and techniques are inadequate, in general [16], explained as follows.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 50–60, 2008.

Consider an evaluation of a probabilistic logic program in which we obtain two derivations of a ground atom A with identical certainty values, say 0.5. We denote this by two copies of atom-certainty pairs A:0.5. Suppose these two derivations are independent, e.g., obtained by applying two "independent" rules in the program. Then, the overall certainty of A would be 0.75, obtained by combining multiple derivations into a single certainty using the disjunction function $ind(\alpha,\beta)=\alpha+\beta-\alpha\beta$, for probabilistic independence mode. Had we considered this collection of derivations as a set, we would get 0.5 as the certainty associated with A – an incorrect result.

We introduced the *parametric framework* [10], a language which generalizes and unifies a class of proposed frameworks with uncertainty. In [18], we studied challenges in bottom-up, fixpoint evaluation of programs in the parametric framework over the certainty domain [0,1] and developed a semi-naïve (SN) method. The considerable improvement achieved by SN over the naive method was due to avoiding or reducing unnecessary computations. To be more precise, in the SN method we restrict at iteration *i*, applying only those rules r for which we obtained at iteration *i*-1, a "new" derivation A:α for a subgoal in the body of r. In our context, "new" means either A was derived for the first time or its associated certainty α was improved. For the computation to be correct, a run-time mechanism is required to ensure atom-certainty pairs derived by the same rule but in different iterations are not combined. To implement this in the SN method proposed in [18], we used annotated atoms, e.g., if an atom A is derived at iteration i by applying rule r, then all derivations A:α by r in the multiset $M_i(A)$ associated with A are removed and replaced by new derivations by r. This removing process included also those derivations by r whose certainties were not changed. This is a source of inefficiency addressed in this paper. The following example illustrates this point.

Let r be a rule in a p-program P in the parametric framework:

$$r: p(X,Y) \xleftarrow{\alpha} q(X,Z), t(Z,Y); < f_d, f_p, f_c > .$$

where p and q are IDB predicates (also called intensional database relations), *t* is an extensional relation (EDB), and $<f_d, f_p, f_c>$ is the list of disjunction, propagation, and conjunction functions associated with r. Suppose r has three instances which yield the following derivations of p(1,2) at iteration i, with certainties β_1, β_2, and β_3, resp.

$$d1: p(1,2) \xleftarrow{\alpha} q(1,3), t(3,2); < f_d, f_p, f_c > .$$

$$d2: p(1,2) \xleftarrow{\alpha} q(1,4), t(4,2); < f_d, f_p, f_c > .$$

$$d3: p(1,2) \xleftarrow{\alpha} q(1,5), t(5,2); < f_d, f_p, f_c > .$$

In a SN evaluation, the multiset associated with p(1,2) at iteration i would be $M_i(p(1,2))=\{|r: \beta_1, r: \beta_2, r: \beta_3|\}$. Assume that at the end of iteration i, only the certainty of q(1,5) is increased, say from δ to γ. Thus, the evaluation should proceed to the next iteration, i+1. Before this is done, however, every derivation by r is removed from $M_i(p(1,2))$, since q is used in r. Ideally, we should keep the derivations d1 and d2 of r, and re-evaluate only d3. While existing powerful engines such as XSB [16] or CORAL [14] support multisets, they are not suitable in our context as they allow derivations across iterations to be combined or even perform this to improve efficiency, which may yield incorrect results when the program defines a recursive predicate whose associated disjunction function is e.g., *ind*. Technical details of an

interesting work related to uncertainty is discussed in [11] for annotated logic programming proposed [6] with the set semantics.

We propose a semi-naïve method which performs the required computation with multisets efficiently. It adopts partitioning and backtracking (hence called SNPB) from standard case, which partitions each IDB relation into two parts: *improved* and *unchanged*. The evaluation proceeds by applying only those rules with a subgoal whose improved part was extended by addition of some new tuples. A backtracking feature in SNPB is used to identify and re-evaluate derivations of those tuples which used some tuples moved from the unchanged part to improved part. We implemented a running prototype of SNPB. Our experimental results indicate superiority of SNPB over SN; in some cases by about two order of magnitude. The overhead of the runtime mechanism in SNPB never caused a disadvantage, according to our experiments.

The rest of this paper is organized as follows. Next, we review the parametric framework as a background. In Section 3 we introduce the SNPB method and establish its correctness. Experiments and results are reported in Section 4. Concluding remarks and future directions are provided in Section 5.

2 The Parametric Framework: A Review

On the basis in which certainty values are associated with facts and rules in the proposed frameworks with uncertainty, we classified them into *annotation based* (AB, for short) and *implication based* (IB) approaches [16]. In the former, a certainty is associated with each subgoal in a rule, while in the latter, it is associated with the implication in the rule. The expressive power of these two approaches is studied in [17]. We then proposed a parametric framework for uncertainty [16] with an IB approach which unifies and/or generalizes all the *IB* frameworks, in the sense that every program in an *IB* framework may be expressed as a parametric program (p-program) by proper selecting of the parameters.

Let L be a first order language that contains infinitely many variables, and finitely many constants and predicates, but no function symbols. While L does not contain function symbols, it contains symbols for families of propagation F_p, conjunction F_c, and disjunction functions F_d, to all of which we refer collectively as *combination* functions $F=F_c \cup F_p \cup F_d$. These functions together with the certainty domain form the "parameters" In this paper, we consider the certainty domain C=[0,1], assumed to be a complete lattice with the usual ordering $\leq$, and with *max* as the join operator and *min* as the meet. A p-program is a set of rules, each of which is an expression of the form:

$$H \xleftarrow{\alpha} B_1,...,B_n;\langle f_d, f_p, f_c\rangle,$$

where H, $B_1,\ldots,B_n$ are atomic formulas, α in (0,1] is the rule certainty, f_d is the disjunction associated with the head predicate, f_p is the propagation function associated with r, and f_c is its conjunction function. For an atom A, we use $\pi(A)$ to denote the predicate symbol of A, and use $Disj(\pi(A))$ to denote the disjunction function associated with this predicate. For consistency reason, the disjunction function associated with all rules defining the same predicate required to be identical.

Let P be a p-program, B_P be the Herbrand base of P, and C be the certainty domain. A valuation υ of P is a mapping from B_P to C, which to every ground atom in B_P,

assigns a value in *C*. Let r be a rule in P. A ground instance of r is a rule obtained by replacing all occurrences of each variable in r with a constant in the Herbrand universe. The Herbrand instantiation of a p-program P, denoted as P^*, is the set of all ground rules in P. Extended from the standard case, the fixpoint semantics of p-programs is based on the notion of immediate consequence operator T_P, which is a mapping from γ_P, the set of valuations to γ_P, such that for every $\upsilon \in \gamma_P$ and every ground atom $A \in B_P$, $T_P(\upsilon)(A) = f_d(X)$, where $f_d = Disj(\pi(A))$ and X is the multiset of certainty values $f_p(\alpha_r, f_c(\{|\upsilon(B_1),\ldots,\upsilon(B_n)|\}))$, for every rule in P^* whose head is A, i.e., the rule instance $(A \xleftarrow{\alpha_r} B_1,\ldots,B_n; \langle f_d, f_p, f_c \rangle)$ is in P^*.

The bottom-up evaluation of T_P is then defined as in the standard case. It has been shown that T_P is monotone and continuous, and that for any p-program P, the least fixpoint of T_P, denoted lfp(T_P), coincides with the declarative semantics of P [16]. Note that lfp(T_P) is reached at iteration step in which no atom is derived whose certainty is improved. The role of disjunction functions *f* is crucial in fixpoint computation in our context when $f(\alpha,\beta) > max(\alpha,\beta)$, for $0<\alpha<1$ and $0<\beta<1$. We call such a disjunction function *f* as type 2, and *max* which coincides with the lattice join as type 1 [16]. If a type 2 disjunction is associated with a recursive predicate in a program, it may cause the evaluation not to terminate. A basic semi-naïve (SN) method defined similar to the standard case was explained in our illustrating example.

3 Semi-naive Evaluation with Partitioning and Backtracking

To further improve the performance and reduce repeated computation with multisets, we propose a refinement of the SN method, called semi-naive with partitioning and backtracking (SNPB). There are two main steps. The first step focuses on derivations that may generate improved atom-certainty pairs. This is done by partitioning the IDB relations. The second step identifies and stores tuples which need not to be recomputed in future iterations. This is done through a backtracking technique.

Derivations of new atom-certainty pairs are the results of joining relations in a rule body for which in the lat iteration, we derived at least a new tuple, or improved its certainty. Based on this, in this work we adopt the idea of rule rewriting introduced for bottom-up evaluation of Datalog programs to restrict the evaluation to derivations in which newly derived atoms may contribute. Basically, every IDB relation R is partitioned into two parts: the improved part Δ, and the unchanged part Λ which includes the rest of atom-certainty pairs in R.

Let P be a p-program with the EDB predicates B_i's and the IDB predicates T_j's. A generic rule r in P is as follows, in which each argument is a list of variables.

$$r: S(\overline{Z}) \xleftarrow{\alpha} B_1(\overline{X}_1),\ldots, B_n(\overline{X}_n), T_1(\overline{Y}_1),\ldots, T_m(\overline{Y}_m); < f_d, f_p, f_c > .$$

When partitioned, this rule can be rewritten as the following set of rules:

$$r_1: S(\overline{Z}) \xleftarrow{\alpha} B_1(\overline{X}_1),\ldots, B_n(\overline{X}_n), \Delta T_1(\overline{Y}_1), T_2(\overline{Y}_2),\ldots, T_m(\overline{Y}_m); < f_d, f_p, f_c > .$$

$$\vdots$$

$$r_j: S(\overline{Z}) \xleftarrow{\alpha} B_1(\overline{X}_1),\ldots, B_n(\overline{X}_n), \Lambda T_1(\overline{Y}_1),\ldots, \Lambda T_{j-1}(\overline{Y}_{j-1}), \Delta T_j(\overline{Y}_j), T_{j+1}(\overline{Y}_{j+1}),\ldots, T_m(\overline{Y}_m);$$

$$\vdots$$

$$r_m: \quad S(\overline{Z}) \xleftarrow{\alpha} B_1(\overline{X}_1),...,B_n(\overline{X}_n), \Lambda T_1(\overline{Y}_1),..., \Lambda T_{m-1}(\overline{Y}_{m-1}), \Delta T_m(\overline{Y}_m); < f_d, f_p, f_c >.$$

$$r_{m+1}: \quad S(\overline{Z}) \xleftarrow{\alpha} B_1(\overline{X}_1),...,B_n(\overline{X}_n), \Lambda T_1(\overline{Y}_1),..., \Lambda T_{m-1}(\overline{Y}_{m-1}), \Lambda T_m(\overline{Y}_m); < f_d, f_p, f_c >.$$

It has been shown for Datalog, the de facto standard deductive database language, this rewriting is equivalent to the original rule. Since our framework uses multiset as the semantics structure, we need to establish correctness of this rewriting, and hence the following result. For lack of space, we suppress the proofs of our results.

Theorem 1. *Given any parametric rule, the set of rules generated by the rewriting technique above produces an equivalent rule, under the multiset semantics.*

Note that since the last rule r_{m+1} does not include a join that involves new facts, we can store derivations of such rules and avoid re-evaluating them later. Given a program P, we introduce in Fig. 1 a procedure called $Eval^i$, which evaluates all but the last rewritten rules defining a predicate T at iteration *i.*

Procedure $Eval^i_T(\Lambda T_1^i,\ldots, \Lambda T_m^i, T_1^i,\ldots, T_m^i, \Delta^i_{T1},\ldots,\Delta^i_{Tm})$
  ForAll $r: S(\overline{Z}) \xleftarrow{\alpha} B_1(\overline{X}_1),..., B_n(\overline{X}_n), T_1(\overline{Y}_1),..., T_m(\overline{Y}_m); < f_d, f_p, f_c >$
    in P, where S = T, $T_1,\ldots,T_m$ are the *IDB* and $B_1,\ldots,B_n$ are the *EDB* relations:
1 Rewrite r using the rewriting technique above;
2 Evaluate all the rewritten rules, except the last one, on the input;
  End ForAll;
  Return S which is the set of atom-certainty pairs computed;
End Procedure

Fig. 1. Procedure for evaluation of rewritten rules

Every partition of an IDB relation needs to be updated at the end of each iteration if there is a change in the part Λ. Some facts in Λ may get a better certainty and hence need to be moved to part Δ. Since atoms with improved certainties will be re-evaluated in the next iteration and their results will be added to the bookkeeping later on, their corresponding "old" atom-certainty pairs in the bookkeeping need to be moved out to avoid duplicate counting of the same derivation across different iterations. This brings out the need for a backtracking maintenance in step two.

The problem with updating the bookkeeping is that we do not know which records were derived using the improved facts. For this, we introduce a backtracking method, which essentially applies the evaluation of the rewritten rules. It identifies the tuples that should be re-computed and replaced. For instance, assume that the bookkeeping contains a record α for atom A, and α is derived from fact-certainty pair B:β. Suppose B is improved from β to γ at iteration i. Then all tuples derived from B are removed from the bookkeeping. To do so, we re-compute the derivations in which B is involved. For this, it uses the certainty β of the subgoal B obtained at iteration i-1, and generates the tuple A:α, which is then located and removed from the bookkeeping.

The above steps are formally expressed as the algorithm shown Fig.2. It classifies rules in the input p-program into two partitions: rules with no IDB predicate in the

body, and rules with some IDB predicates in the body. We first evaluate the former type of rules, which can be done in a single iteration and no backtracking is required. The result of this iteration is used to initialize the bookkeeping structure and partitions (lines 1-2). In each iteration step, we add the new evaluation results (line 4) to the bookkeeping. If the certainty of any fact is improved, the system backtracks the results derived using such facts (line 6) and removes them from the bookkeeping (lines 7-9). The repartitioning is done in lines 10 and 11. This process continues until every partition ΔT is empty, in which case the least fixpoint is obtained.

```
    Procedure Semi-Naïve-Partition-Backtracking (P,D; lfp(P∪D))
1       Set P' to be the rules in P with no IDB predicate in the body;
        ΛT^1 := ∅, BK^0_T := ∅, for the IDB predicate T defined in P;
        MΔ^1_T := P'(I)(T), for each IDB predicate T;
        BK^1_T := BK^0_T ∪ MΔ^1_T ;
2   T^1 := f_d(BK^1_T) ; Δ^1_T := f_d(MΔ^1_T);  i := 1;
      Repeat
        ForAll IDB predicate T, where T_1,….,T_m are the IDB
              predicates used in the definition of T:
3                 MΔ^{i+1}_T := Eval^i_T(ΛT_1^i,…, ΛT_m^i,T_1^i,… T_m^i,Δ^i_T1,…,Δ^i_Tm);
4                 BK^{i+1}_T := BK^i_T ∪ MΔ^{i+1}_T;
5                 T^{i+1} := f_d (BK^{i+1}_T);
6                 Change^i_T :={(T(μ),C_i) | (T(μ),C_{i+1})∈T^{i+1} ∧ (T(μ),C_i)∈T^i ∧ C_{i+1}> C_i}
7                 ΛT^i := T^i - Change^i_T ;  DΔ^i_T1:= Change^i_T ;
8                 Remove^{i+1}_T := Eval^i_T(ΛT_1^i,…, ΛT_n^i, T_1^i,… T_n^i, DΔ^i_T1 …DΔ^i_Tn)
9                 BK^{i+1}_T := BK^{i+1}_T - Remove^{i+1}_T ;
10                ΛT^{i+1} := T^i - Change^i_T;
11                Δ^{i+1}_T := T^{i+1} - ΛT^{i+1} ;
            End ForAll;
12      i := i +1;
        Until Δ^i_T = ∅, for each IDB predicate T;
        lfp(P∪D) := ⋃T^i;
    End Procedure;
```

Fig. 2. The semi-naive with partitioning and backtracking algorithm (SNPB)

Our next result establishes correctness of the SNPB algorithm we proposed.

Theorem 2. *Let P be any p-program and D be an input dataset of atom-certainty pairs. A fixpoint evaluation of P on D using the semi-naïve with partitioning and backtracking algorithm produces the same result as the naive method.*

4 Experiments and Results

To measure performance of the SNPB algorithm, we developed a prototype in C++ and conducted experiments using benchmark data we generated of different structures and sizes. We measured the elapsed time taken from submitting a query to producing the last answer tuple. The computer system we used for these experiments was a typical IBM desktop with a Pentium 4 CPU of 2.4GHz, 512MB main memory, 80GB hard disk, running Windows 2000. We used the standard memory block size.

$r_1 : p(X, Y) \xleftarrow{1} e(X, Y) ; \langle ind, min, - \rangle.$ $r_2 : p(X, Y) \xleftarrow{1} e(X, Z), p(Z, Y); \langle ind, min, min \rangle.$

(a) Test program *P1*

$r_1 : p(X, Y) \xleftarrow{1} e(X, Y) ; \langle ind, min, - \rangle.$ $r_2 : p(X, Y) \xleftarrow{1} p(X, Z), p(Z, Y); \langle ind, min, min \rangle.$

(b) Test program P2

$r_1 : sg(X, Y) \xleftarrow{1} flat(X, Y) ; \langle ind, *, min \rangle.$

$r_2 : sg(X, Y) \xleftarrow{1} up(X, Z1), sg(Z1, Z2), flat(Z2, Z3), sg(Z3, Z4), down(Z4, Y); \langle ind, *, min \rangle.$

(c) Test program P3

$r_1 : msg(1) \xleftarrow{1} ; \langle ind, -, - \rangle.$

$r_2 : supm2(X,Y) \xleftarrow{1} msg(X), up(X, Y); \langle ind, *, min \rangle.$

$r_3 : supm3(X, Y) \xleftarrow{1} supm2(X, Z), sg(Z, Y); \langle ind, *, min \rangle.$

$r_4 : supm4(X, Y) \xleftarrow{1} supm3(X, Z), flat(Z, Y); \langle ind.*, min \rangle.$

$r_5 : sg(X, Y) \xleftarrow{1} msg(X), flat(X, Y); \langle ind, *, min \rangle.$

$r_6 : sg(X, Y) \xleftarrow{1} supm4(X, Z), sg(Z, W), down(W, Y); \langle ind, *, min \rangle.$

$r_7 : msg(X) \xleftarrow{1} supm2(Z, X); \langle ind, *, - \rangle.$

$r_8 : msg(X) \xleftarrow{1} supm4(Z, X); \langle ind, *, - \rangle.$

(d) Test program P4

Fig. 3. Test programs used in our experiments

As for the test programs, we considered four classes of p-programs *P1* to *P4* shown in Fig. 3, obtained by extending from standard programs used by others for performance evaluation The test programs *P1* and *P2* compute transitive closure, and *P3* and *p4* are the so-called same-generation programs. While *P3* contains a rule with many subgoals, *P4* contains a number of rules with fewer subgoals. In fact, *P4* is the same as *P3* but extended with supplement magic predicates and rules [2]. Also, we used the disjunction function *ind* for every IDB predicate in these programs.

As for the EDB, we used 9 datasets of different sizes and structures, shown in the appendix. Of these, CT_n and $M_{n,m}$ are used for programs *P1* and *P2*, where CT_n contains a single cycle and $M_{n,m}$ contains multi-nested cycles. As a result, the number of paths between two nodes (therefore the number of joins) in $M_{n,m}$ is much larger than in CT_n, indicating more workload at every evaluation step for $M_{n,m}$. The other datasets, A_n, B_n, C_n, F_n, S_n, $T_{n,m}$, and $U_{n,m}$, are made suitable and used for *P3* and *P4*. These datasets were originally proposed to evaluate standard Datalog programs [7], [15], which we modified by assigning certainty values to EDB facts.

The partial experimental results are presented in Figs. 4 to 7. In these graphs, the x axis represents the number of EDB tuples and the y axis represents the elapsed time in milliseconds. The curves marked as “Naive”, “SN”, and “SNPB” represent the elapsed time, given different numbers of EDB data, of the evaluation using Naive, SN, and SNPB techniques, respectively. We define the speed-up as the ratio of the evaluation time between two techniques.

Our results show that, keeping the same number of iterations to reach the fixpoint, SNPB is faster than Naïve and SN. Moreover, when the EDB size grows, the speed-up gained by SNPB increases rapidly. For instance, let us consider (see Fig. 6). The speed up obtained by SNPB compared to Naïve method grows from 1.5 to 437.38, and for SN compared to Naïve, the speed up ranges from 1.25 to 203.26, when the data layers in A_n increase from 4 to 9 (EDB size from 58 tuples to 2042). This indicates the larger the size of the input dataset, the higher efficiency SNPB provides. This also indicates scalability of SNPB, which is important for handling large EDB datasets.

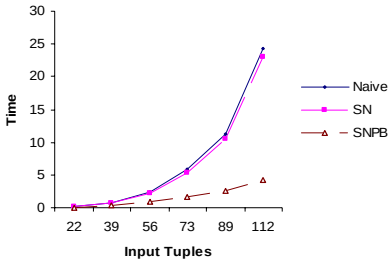


Fig. 4. Evaluation of *p1* on the dataset $M_{n,m}$

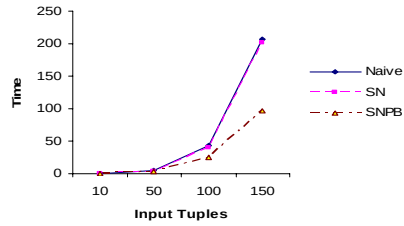


Fig. 5. Evaluation of *p2* on the dataset CT_n

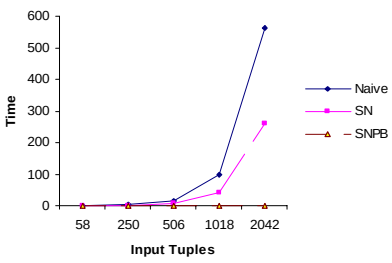


Fig. 6. Evaluation of *p3* on the dataset A_n

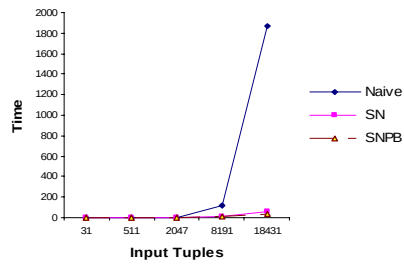


Fig. 7. Evaluation of *p4* on the dataset S_n

Looking at the results of evaluation of *P1* and *P2*, we can see that the number of iterations to reach the fixpoint for *P1* is more than *P2* (e.g. 102 vs. 9 for *P1* on $M_{100,4}$ vs. *P2* on CT_{100}). This is because the linear rule in *P1* generates fewer new IDB facts at each iteration compared to the non-linear rule in *P2*. Moreover, $M_{n,m}$ provides more paths between two nodes than CT_n does. By applying *ind* as disjunction function, the certainties of transitive pairs in $M_{n,m}$ will have more chances to grow, compared to those in CT_n. This results in increased number of steps for evaluating *P1* on $M_{n,m}$.

By running *P3* on A_n, there are fewer facts generated at each iteration, on average, by the recursive rule and the size of the recursive relation is relatively large. In this situation, many repeated derivations performed by SN are detected and avoided by SNPB. Since the derivations by the recursive rule involve many joins, the speed-up of SNPB over SN is more (see Fig. 6).

Fig. 7 shows the results of running *P4* on S_n. Unlike the gain obtained when running *P3* on A_n, SNPB is slightly better than SN. If we examine *P4* more carefully, we note that there are some rules in *P4* which are only evaluated at some particular iterations in both SN and SNPB methods. This is because at each iteration we may derive a new fact useful to some but not all the rules. The saving from avoiding useless evaluations of rules reduces the evaluation time for both SN and SNPB. Moreover, the size of each IDB relation and the number of related facts derived during the evaluation are also small. These lead to less number of repeated derivations, which could be avoided in SNPB, and hence little saving by SNPB. For the input program S_{96}, the efficiency observed by SNPB over SN is twice.

To summarize, SNPB revealed significant increased efficiency compared to naive and SN methods in most cases. Our results also indicated that the speed-up gained by SNPB over SN varies, depending on the program structure and the size and structure of the dataset considered. When SN provided significant performance compared to the naive method, e.g. running *P4*, there is more room for SNPB to exploit for improved efficiency. In cases where SN performs many repeated computations, e.g., running programs *P1*, *P2*, and *P3*, the SNPB method indicated superiority over SN.

5 Conclusions and Future Work

We studied bottom-up evaluation of logic programs with uncertainty over the domain [0, 1]. We showed that presence of type 2 disjunction functions pose a challenge to develop a run-time mechanism which implements the SN method while ensures derivations across different iterations and not combined. We proposed such a method which uses partitioning and backtracking, and established its correctness. The efficiency of the proposed algorithm SNPB was shown through our numerous experiments indicating its superiority over the basic SN method. Our results also indicate scalability of SNPB, important for handling larger datasets.

Extending SNPB to a disk-based model is important for persistency. We are also investigating ways to extend our work with magic sets, a rewriting technique for Datalog which takes into account the query structure during a fixpoint evaluation.

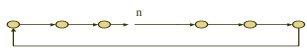
Acknowledgments. This work was supported in part by Natural Sciences and Engineering Research Council (NSERC) of Canada and by Concordia University.

References

1. Abiteboul, S., et al.: The Lowell Database Research Self-Assessment. Communication of the ACM 48(15), 111–118 (2005)
2. Beeri, C., Ramakrishnan, R.: On the Power of Magic. J. of Logic Programming 10, 255–299 (1991)
3. Ceri, S., Gottlob, G., Tanca, L.: What You Always Wanted to Know About Datalog (and Never Dared to Ask). IEEE TKDE 1(1), 146–166 (1989)
4. Dubois, D., Lang, J., Prade, H.: Towards Possibilistic Logic Programming. In: 10th International Conference on Logic Programming, pp. 581–595 (1991)

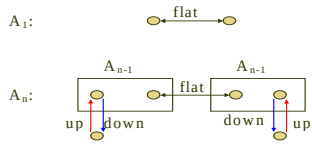
5. Kifer, M., Li, A.: On the Semantics of Rule-Based Expert Systems with Uncertainty. In: 2nd ICDT, pp. 102–117 (1988)
6. Kifer, M., Subrahmanian, V.S.: Theory of Generalized Annotated Logic Programming and its Applications. Journal of Logic Programming 12(3&4), 335–367 (1992)
7. Kuittinen, J., Nurmi, O., Sippu, S., Soininen, E.S.: Efficient Implementation of Loops in Bottom-up Evaluations of Logic Queries. In: 16th International Conference on Very Large Data Bases Conference, Queensland, Australia, pp. 372–379 (1990)
8. Lakshmanan, L.V.S., Sadri, F.: Probabilistic Deductive Databases. In: Symp. on Logic Programming, pp. 254–268 (1994)
9. Lakshmanan, L.V.S., Sauri, F.: Modeling Uncertainty in Deductive Databases. In: Karagiannis, D. (ed.) DEXA 1994. LNCS, vol. 856, pp. 724–733. Springer, Heidelberg (1994)
10. Lakshmanan, L.V.S., Shiri, N.: A Parametric Approach to Deductive Databases with Uncertainty. IEEE TKDE 13, 554–574 (2001)
11. Leach, S.M., Lu, J.J.: Query Processing in Annotated Logic Programming: Theory and Implementation. Journal of Intelligent Information Systems 6(1), 33–58 (1996)
12. Lloyd, J.W.: Foundations of Logic Programming, 2nd edn. Springer, Heidelberg (1987)
13. Ng, R.T., Subrahmanian, V.S.: Probabilistic Logic Programming. Information and Computation 101(2), 150–201 (1992)
14. Ramakrishnan, R., Srivastava, D., Sudarshan, S., Seshadri, P.: The CORAL Deductive System. VLDB Journal 3(2), 161–210 (1994)
15. Ramakrishnan, R., Ullman, J.D.: A Survey of Deductive Database Systems. Journal of Logic Programming 23(2), 125–149 (1995)
16. Sagonas, K.F., Swift, T., Warren, D.S.: XSB as an Efficient Deductive Database Engine. In: SIGMOD Conference, pp. 442–453 (1994)
17. Shiri, N.: Expressive Power of Logic Frameworks with Uncertainty. In: 18th Int'l FLAIRS Conference, Special Track on Uncertainty Reasoning, Florida, May 16–18 (2005)
18. Shiri, N., Zheng, Z.: Challenges in Fixpoint Computation with Multisets. In: International Symposium on Foundations of Information and Knowledge Systems, pp. 273–290 (2004)
19. Van Emden, M.H.: Quantitative Deduction and Its Fixpoint Theory. Journal of Logic Programming 4, 37–53 (1986)

Appendix: Benchmark Datasets

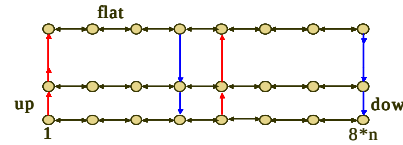


(a) T_n

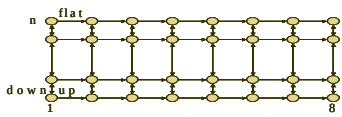
(b) M_n



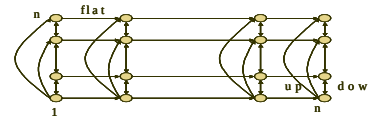
(c) A_n



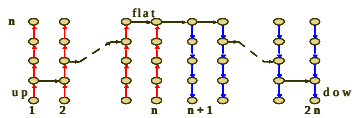
(d) B_n



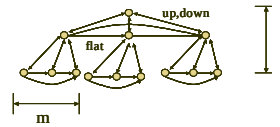
(e) C_n



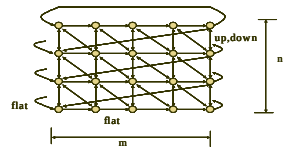
(f) F_n



(g) S_n



(h) $T_{n,m}$



(i) $U_{n,m}$

Approximating Component Selection with General Costs

Mostafa Nouri and Jafar Habibi

Computer Engineering Department,
Sharif University of Technology, Tehran, Iran
nourybay@ce.sharif.edu, jhabibi@sharif.edu

Abstract. In the past decades there has been a burst of activity to simplify implementation of complex software systems. The solution framework in software engineering community for this problem is called component-based software design (CBSD), whereas in modeling and simulation community it is called composability. Composability is a complex feature due to the challenges of creating components, selecting combinations of components, and integrating the selected components.

In this paper we address the second challenge through the analysis of Component Selection (CS), the NP-complete process of selecting a minimal set of components to satisfy a set of objectives. Due to the computational complexity of CS, we examine approximating solutions that make the CS process practical. We define three variations of CS and present good approximation algorithms to find near optimal solutions. In spite of our creation of approximable variations of Component Selection, we prove that the general Component Selection problem is inapproximable.

Keywords: Component Selection, NP-Completeness, Set Cover, Approximation Algorithms, Red-Blue Set Cover.

1 Introduction

A recent topic of interest in the software engineering community and the modeling and simulation community involves the notion of component-based software development. There had been many efforts in the software community to incorporate the concept of reusing. Components offer a useful mechanism to support reuse. But a number of questions are raised by them as well.

Composability is the ability to combine reusable simulation components to satisfy a set of user objectives. Composability is a highly demanded goal for model and simulation developers because of the benefits afforded by reuse. Component Selection (CS) is the problem of choosing the minimum number of components from a set of components such that their composition satisfies a set of objectives. CS which is an NP-complete [5] optimization problem formally shown to be embedded within composability [5]. For composability to be achievable in reasonable time, the selection of a set of components must be accomplishable in polynomial-time. Therefore we should try to find a polynomial-time algorithm

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 61–68, 2008.

to solve CS Problem. In [2] it has been conjectured that in the general case CS is inapproximable.

In this paper we focus on Component Selection problem and try to solve it approximately in different cases. The results we have obtained can be summarized as below:

- Prove that Component Selection problem is inapproximable in the general case, a conjecture which is suggested by Fox *et al.* in [2].
- Give an approximation algorithm for Component Selection problem, when components have unit costs and compositions have not unbounded emergent and non-emergent behaviors. We also give the upper bound of the approximation ratio.
- Define two new version of Component Selection problem, with real or general costs, and offer the approximation algorithms and the upper bound of the approximation ratios.

In the following we will have examine component selection in three different cases. In section 2 we will study the well-known problem of component selection with unit cost. Then in section 3 we will extend the problem to the case that components have some real valued costs. In section 4 the general component selection are defined and the approximation algorithm for it is presented. Finally in section 5 we will summarize the results of this paper.

2 Component Selection with Unit Costs

In this section we consider the simplest form of component selection, CSUC, in which adding each component poses a unit cost to the composition. Informally the problem is to select and compose minimum number of components from a repository of components such that the composition will meet a given set of objectives.

Let $O = \{o_1, o_2, ..., o_n\}$ be a set of objectives and $C = \{c_1, c_2, ..., c_m\}$ be a set of components. A simulation system S is a subset of C, i.e. $S \subseteq C$. if $|S| > 1$ then S is a composition. Let $\circ$ denote composition of components, e.g. $(c \circ c')$ is the composition of c and c'. Let $\vDash$ and $\nvDash$ denote *satisfying* and *not satisfying* an objective respectively, e.g. $c \vDash o$ means component c satisfies objective o and $c \nvDash o$ means component c does not satisfy objective o. These two operators may also be used with compositions and sets of objectives and have the expected meanings, e.g. $c \circ c' \vDash o$ and $S \nvDash O$. A simulation system $S \vDash O$ if and only if for every $o_i \in O$, $S \vDash o_i$.

In some variants of CSUC, the set of objectives satisfied by a composition may not be the union of the objectives satisfied by the components individually. With regard to the set of satisfied objectives by a composition of a set of components there may be three different situations which can be seen in Table 1. Two of these situation (emergent and non-emergent) were introduced by Page and Opper [3] and the third one (anti-emergent) was later defined by Petty *et al.* [5]. Informally if none of c and c' and $c \circ c'$ satisfy o, then the composition is *non-emergent*. If c

Table 1. Different types of compositions

Objective Satisfaction of Components		Objective Satisfaction of Composition	Composition Type
$c \vDash o$	$c' \vDash o$	$(c \circ c') \vDash o$	Non-emergent
$c \nvDash o$	$c' \vDash o$	$(c \circ c') \vDash o$	Non-emergent
$c \vDash o$	$c' \nvDash o$	$(c \circ c') \vDash o$	Non-emergent
$c \nvDash o$	$c' \nvDash o$	$(c \circ c') \nvDash o$	Non-emergent
$c \nvDash o$	$c' \nvDash o$	$(c \circ c') \vDash o$	Emergent
$c \vDash o$	$c' \vDash o$	$(c \circ c') \nvDash o$	Anti-emergent
$c \nvDash o$	$c' \vDash o$	$(c \circ c') \nvDash o$	Anti-emergent
$c \vDash o$	$c' \nvDash o$	$(c \circ c') \nvDash o$	Anti-emergent

and c' do not satisfy o but $c \circ c'$ satisfy o, then the composition is *emergent*. If one or both of c and c' satisfy o but $c \circ c'$ does not satisfy o, then the composition is *anti-emergent*. In Table 1 all different possible logical combinations of satisfaction of an objective by components have been shown.

Petty *et al.* [5] defined a general problem which subsumes all these different variations of the problem. They showed that even in the presence of an oracle function that can determine in one step which objectives are satisfied by a component or a composition, the problem of component selection with unit costs (CSUC) is NP-complete. The formal definition of the decision version of CSUC is as follows:

> *Instance*: Set $C = \{c_1, c_2, ..., c_m\}$ of components, set $O = \{o_1, o_2, ..., o_n\}$ of objectives, oracle function $\sigma : 2^C \to 2^O$, positive integer $K \leq |C|$.
> *Question*: Does C contain a composition S that satisfies O of size K or less, i.e. a subset $S \subseteq C$ with $|S| \leq K$ such that $O \subseteq \sigma(S)$?

Since this problem has been proved to be NP-complete, it is very unlikely to find a polynomial time algorithm for solving CSUC (unless P=NP). Therefore it is natural to seek an algorithm that approximately computes the minimum number of components needed to satisfy all the objectives in O.

Fox *et al.* [2] have demonstrated that the greedy algorithm is not an approximation algorithm for CSUC. Since the greedy algorithm is one of the best algorithm for Minimum Set Cover problem which is related to the CSUC, they conjectured that there is no algorithm that can approximate the CSUC problem. We here claim that in the general case, the CSUC problem cannot be approximated.

Theorem 1. *There is no approximation algorithm for CSUC when the compositions in CSUC may have some emergent or anti-emergent rules.*

Proof. The proof is in the full version of the paper.

While the general CSUC problem is not approximable, we can use the greedy algorithm as it is used for Minimum Set Cover problem and get an approximation ratio of $H(k) = \sum_{i=1}^{k} \frac{1}{i}$ where $k = \max\{|\sigma(c)| : c \in C\}$, when the composition only exhibits non-emergent behavior. Also when the number of emergent and anti-emergent behaviors are polynomially bounded and can be iterated by σ, we can use a modified version of the greedy algorithm and get the same ratio. The greedy algorithm when CSUC only exhibits non-emergent behavior can be seen in Algorithm 1. In the following theorem it will be proved that the approximation ratio of the algorithm is $H(k)$.

Algorithm 1. The algorithm for CSUC problem with only non-emergent rules.

```
1: function GREEDY-CSUC(C, O, σ)
2:     U ← O                                  ▷ To be satisfied objectives
3:     S ← ∅                                  ▷ Selected components
4:     while U ≠ ∅ do                         ▷ Some unsatisfied objective
5:         Select a c_i ∈ C that maximizes |σ(c_i) ∩ U|
6:         U ← U − σ(c_i)                     ▷ Remove satisfied objectives
7:         S ← S ∪ {c_i}                      ▷ Select component
8:         return S
9:     end while
10: end function
```

Theorem 2. *The algorithm* GREEDY-CSUC *when σ has no emergent or anti-emergent behavior, approximates the solution with ratio $H(k) = \sum_{i=1}^{k} \frac{1}{i}$ where $k = \max\{|\sigma(c)| : c \in C\}$.*

Proof. The proof is in the full version of the paper.

3 Component Selection with Real Costs

In this section we extend the previous problem of component selection by assigning a real number as a cost to each component or composition, therefore we call it Component Selection with Real Costs (CSRC). This problem has more application than the previous one. For example suppose that a few components satisfy most of the objectives of the problem, but they need many modification and integration efforts. But we can also use some more cheap components to satisfy the same objectives. The problem in this case is "what should we do?". Should we select few costly components or many inexpensive components? This problem can easily been modeled by CSRC. The formal definition of the decision version of CSRC is as follows:

Instance: Set $C = \{c_1, c_2, ..., c_m\}$ of components, set $O = \{o_1, o_2, ..., o_n\}$ of objectives, oracle function $\sigma : 2^C \to 2^O$, cost function $\omega : 2^C \to \mathbb{R}^+$, positive integer $K \leq |C|$.
Question: Does C contain a composition S that satisfies O and its cost is K or less, i.e. a subset $S \subseteq C$ such that $\omega(S) \leq K$ and $O \subseteq \sigma(S)$?

CSRC is as hard as the CSUC, since it contains the CSUC as a special case. When the cost of each composition is equal to the number of components it contains, i.e. $\omega(S) = |S|$, the problem changes into CSUC. So we can immediately get the following theorem.

Theorem 3. *Component selection with real costs (CSRC) is NP-complete.*

As the previous problem, because the problem is NP-complete we should seek an approximation algorithm for CSRC. CSRC is also at least as hard as CSUC, and CSUC cannot be approximated with ratio better than $O(\log n)$ unless P=NP (since it is as hard as minimum set cover problem), so we should not expect a better algorithm than CSUC.

CSRC problem add a new function to CSUC. This function, like σ, can exhibits three different behaviors. These behaviors are *cumulative*, *convergent* and *divergent*. The descriptions are easy, when the cost of a composition is equivalent to the sum of the costs of its components the composition is cumulative. When its cost is less than that value, the composition is convergent and when it is greater, the composition is divergent.

It can be proved, like emergent and anti-emergent behaviors, that when CSRC problem has unbounded number of convergent or divergent compositions, it cannot be approximated. In contrast to these behaviors, when CSRC contains only cumulative composition, it can be approximated with a similar ratio. The algorithm is a modified version of greedy algorithm on the CSUC. The algorithm can be seen in the Algorithm 2.

Algorithm 2. The algorithm for CSRC problem with non-emergent and cumulative behavior.

```
1: function GREEDY-CSRC(C, O, σ, ω)
2:     U ← O                                   ▷ To be satisfied objectives
3:     S ← ∅                                   ▷ Selected components
4:     while U ≠ ∅ do                          ▷ Some unsatisfied objective
5:         Find the most cost effective component,
                   i.e. c_i ∈ C that minimizes ω(c_i) / |σ(c_i) ∩ U|
6:         U ← U − σ(c_i)                      ▷ Remove satisfied objectives
7:         S ← S ∪ {c_i}                       ▷ Select component
8:         return S
9:     end while
10: end function
```

Theorem 4. *The* GREEDY-CSRC *when σ has no emergent or anti-emergent and ω has no convergent or divergent behavior, approximates the solution with ratio $H(n) = \sum_{i=1}^{n} \frac{1}{i}$ where $n = |O|$.*

Proof. The proof is in the full version of the paper.

4 Component Selection with Multi-costs

In this section we extend the component selection problem further. The problem is extended by adding a more general cost to components and compositions. Therefore we call it Component Selection with Multi-Costs (CSMC). In this problem we define some known costs, and assign to each components and/or compositions a subset of these costs. The problem in this case is to make a composition such that the minimum number of these costs are selected. The formal definition of the decision version of this problem is as follows:

> *Instance*: Set $C = \{c_1, c_2, ..., c_m\}$ of components, set $O = \{o_1, o_2, ..., o_n\}$ of objectives, set $D = \{d_1, d_2, ..., d_p\}$ of costs, oracle function $\sigma : 2^C \rightarrow 2^O$, cost function $\omega : 2^C \rightarrow 2^D$, weight function for costs $wd : D \rightarrow \mathbb{R}^+$, positive integer $K \leq |C|$.
> *Question*: Does C contain a composition S that satisfies O and the total weight of its costs is K or less, i.e. a subset $S \subseteq C$ such that $\sum_{d \in \omega(S)} wd(d) \leq K$ and $O \subseteq \sigma(S)$?

In this problem, ω has been changed. In CSRC, ω will return a real number, specifying the cost of using that composition. But in CSMC, ω will return a subset of costs in D and the goal is to minimize the total weight of cost set for the total composition. This problem has two difference with CSRC. First, the compositions in CSRC has only one type of cost, which can be specified by a real number, whereas in CSMC components/compositions may have several types of costs. Second, some components/compositions in CSMC may have common costs, which means if we select a component/composition, we can select the other components/compositions without paying that cost twice. Consequently CSMC is a more powerful problem and can be used to model more complicated cases.

In view of complexities of the two problems, CSMC is at least as hard as *approximable* CSRC. To prove this claim we should convert each instance of CSRC to CSMC problem. Let $P = (C, O, \sigma, \omega)$ be an instance of CSRC. We need to create $P' = (C', O', D', \sigma', \omega', wd')$ an instance of CSMC. Let $C' = C$ and $O' = O$ and $\sigma' = \sigma$. For each component $c_i \in C$ add a cost d_{c_i} to D' and assign it to $\omega'(c_i)$ and set its weight $wd(d_{c_i})$ to $\omega(c_i)$. If ω has convergent or divergent behavior, and the convergent and divergent compositions are polynomially bounded and can be iterated by σ (P is approximable), add corresponding compositions to C' and new costs to D' and assign these costs to the corresponding compositions likewise. Now if P' has an optimum solution with cost w, then there is a $S' \subseteq C'$ such that $O' \subseteq \sigma(S')$ and $\sum_{d \in \omega(S')} wd(d) = w$, for the corresponding subset S' in the problem P, also $O \subseteq \sigma(S)$ (since C, O and σ are equal in both problems) and $\omega(S) = w$ (since the costs of components S' are equivalent to the costs of components of S. Therefore we can conclude the following theorem.

Theorem 5. *CSMC problem cannot be approximated by a ratio better than* $H(n) = \sum_{i=1}^{n} \frac{1}{i}$ *where* $n = |O|$.

We can prove a stronger claim about the approximation ratio of CSMC problem. This ratio comes from the "Red Blue Set Cover" problem. The definition of the decision version of this problem is as follows:

> *Instance*: Let $R = \{r_1, r_2, ..., r_\rho\}$ and $B = \{b_1, b_2, ..., b_\beta\}$ be two disjoint finite sets, and let $\mathcal{S} \subseteq 2^{R \cup B}$ be a family of subsets of $R \cup B$, and let $K \leq |S|$ be an integer.
> *Question*: Does $\mathcal{S}$ contains a subfamily $\mathcal{C} \subseteq \mathcal{S}$ that covers all elements of B, but covers at most K elements of R.

Theorem 6. *CSMC is as hard as RBSC problem.*

Proof. The proof is in the full version of the paper.

Algorithm 3. The approximation algorithm for CSMC problem.

function GREEDY-CSMC($C, O, D, \sigma, \omega, wd$)
  Modify CSMC instance into an instance $\mathcal{T}$ of CSRC as follows:
    $C_\mathcal{T} \leftarrow C, \quad O_\mathcal{T} \leftarrow O, \quad \sigma_\mathcal{T} \leftarrow \sigma,$
    $\omega_\mathcal{T} : 2^C \rightarrow \mathbb{R}^+$ such that $\forall c_i \in C$, $\omega_\mathcal{T}(c_i) \leftarrow \sum_{d_j \in \omega(c_i)} wd(d_j)$.
  return GREEDY-CSRC($C_\mathcal{T}, O_\mathcal{T}, \sigma_\mathcal{T}, \omega_\mathcal{T}$).
end function
function BAD-COSTS($C, O, D, \sigma, \omega, wd, X$)
  Discard components with total cost more than X, i.e.
    $C_X \leftarrow \{c_i \in C | \sum_{d_j \in \omega(c_i)} wd(d_j) \leq X\}$.
  If $O \nsubseteq \sigma(C_X)$ **return** C. ▷ C_X is not feasible
  $Y \leftarrow \sqrt{\frac{n}{\log \beta}}$.
  Separate the costs to *good* and *bad* costs:
    $D_G \leftarrow \{d_j \in D | \sum_{c_i : d_j \in \omega(c_i)} 1 > Y\}, \quad D_B \leftarrow D - D_G$.
  Discard the costs in D_G and set
    $\omega_{X,Y} = 2^C \rightarrow 2^D_G$ such that for all $c_i \in C_X$, $\omega_{X,Y}(c_i) \leftarrow \omega(c_i) - D_G$.
  return GREEDY-CSMC($C_X, O, D_B, \sigma, \omega_{X,Y}, wd$).
end function
function APPROX-CSMC($C, O, D, \sigma, \omega, wd$))
  for $X \leftarrow 1$ to ρ **do**
    call BAD-COSTS($C, O, D, \sigma, \omega, wd, X$).
    Compute the weight of selecting the returned components.
  end for
  return the solution with the least cost.
end function

This problem has been proved to be NP-complete [1]. It has also been proved that this problem cannot be approximated with ratio $2^{log^{1-\epsilon} n}$ for any $0 < \epsilon < 1$ under some plausible complexity theoretic assumptions (such as P $\neq$ NP or NP $\nsubseteq$ DTIME($n^{O(\text{polylog } n)}$). We here show that CSMC is as hard as Red-Blue Set Cover (RBSC) and hence the same bound on its approximation ratio arises.

In spite of the lower bound for approximation ratio of RBSC problem, the best known algorithm for approximating the solution has the ratio $2\sqrt{n \log \beta}$

where $n = |\mathcal{S}|$ and $\beta = |B|$. This algorithm was introduced by Peleg [4]. We can use his algorithm for CSMC problem by transforming each instance of CSMC into an instance of RBSC and then applying the algorithm on this problem. After solving RBSC problem, we can transform the solution backward to find the approximate solution of CSMC. In this section, instead we modify Peleg' algorithm [4] and make it suitable for CSMC problem. The details of greedy algorithm for CSMC can be seen in Algorithm 3.

Theorem 7. *The approximation ratio of the found solution with the optimum solution of CSMC problem is at most* $2\sqrt{n \log \beta}$.

Proof. The proof is in the full version of the paper.

5 Conclusion

Composability is the ability to combine reusable simulation components to satisfy a set of user objectives. Composability is a highly demanded goal for model and simulation developers because of the benefits afforded by reuse. Component Selection (CS) is an NP-complete optimization problem formally shown to be embedded within composability.

The first main result of this paper is that CS in the general formulation is not approximable. Another result of the paper is the definition of three modified versions of CS along with their approximation algorithms. In the first problem each component has a unit cost. In the second problem, each component has a real number as its cost and in the third (and the most general) problem each component may have several types of costs with different weights, and also each component may have some costs common with other components.

For future works we want to seek algorithms with better approximation ratios for these problems (specially for CSMC problem, which there is a gap between the proved upper bound and lower bound). Also we will study other modified versions of Component selection which are tractable by polynomial algorithms.

References

1. Carr, R.D., Doddi, S., Konjevod, G., Marathe, M.V.: On the red-blue set cover problem. In: SODA, pp. 345–353 (2000)
2. Fox, M.R., Brogan, D.C., Reynolde Jr., P.F.: Approximating component selection. In: Winter Simulation Conference, pp. 429–434 (2004)
3. Page, E.H., Opper, J.M.: Observations on the complexity of composable simulation. In: Winter Simulation Conference, pp. 553–560 (1999)
4. Peleg, D.: Approximation algorithms for the label-cover$_{\max}$ and red-blue set cover problems. J. Discrete Algorithms 5(1), 55–64 (2007)
5. Petty, M.D., Weisel, E.W., Mielke, R.R.: Computational complexity of selecting components for composition. In: Fall 2003 Simulation Interoperability Workshop, pp. 553–560 (2003)

Online Suffix Tree Construction for Streaming Sequences

Giyasettin Ozcan and Adil Alpkocak

Dokuz Eylul University, Department of Computer Engineering,
Tinaztepe Buca 35160, Izmir, Turkey
giyaseddin.ozcan@deu.edu.tr, alpkocak@cs.deu.edu.tr

Abstract. In this study, we present an online suffix tree construction approach where multiple sequences are indexed by a single suffix tree. Due to the poor memory locality and high space consumption, online suffix tree construction on disk is a striving process. Even more, performance of the construction suffers when alphabet size is large. In order to overcome these difficulties, first, we present a space efficient node representation approach to be used in Ukkonen suffix tree construction algorithm. Next, we show that performance can be increased through incorporating semantic knowledge such as utilizing the frequently used letters of an alphabet. In particular, we estimate the frequently accessed nodes of the tree and introduce a sequence insertion strategy into the tree. As a result, we can speed up accessing to the frequently accessed nodes. Finally, we analyze the contribution of buffering strategies and page sizes on performance and perform detailed tests. We run a series of experimentation under various buffering strategies and page sizes. Experimental results showed that our approach outperforms existing ones.

Keywords: Suffix trees, sequence databases, time series indexing, poor memory locality.

1 Introduction

Suffix trees, are versatile data structures which enable fast pattern search on large sequences. The large sequence, data set, can be a DNA sequence of a human, whose length is 3 billion; or it can be a collection of musical fragments, where number of fragments in the collection is large but average length of each fragment is moderate[15]. In both sequence cases, total size of the data set may be extremely large. For such large sequence sets, suffix trees introduce a fundamental advantage; sequence search time does not depend on the length of the data set.

The concept of suffix tree construction was initiated before the seventies by a brute force approach [11]. For each suffix insertion, brute force approach preceded a common prefix search operation in the tree. Nevertheless, it was not practical since computational cost of the brute force suffix tree construction was at least exponential. In the seventies, linear time suffix tree construction algorithms were introduced using suffix links and tested on memory. [18,28]. In these algorithms, suffix links functioned as shortcuts, which enable fast access to the suffix insertion positions of the tree. In other words, they hold the address of a node, which contributes to the next suffix insertion position. As a result, traversing the tree for each suffix insertion was

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 69–81, 2008.

not necessary; instead suffix links from the previous step supplied the direct address. Due to this strong advantage, linear time suffix tree construction became possible. [11].

Although early suffix tree construction algorithms ensure linear time construction, they share a common pitfall: the offline property. For instance, in McCreight algorithm [18], all letters of the sequence should be scanned before suffix tree construction procedure starts up. Such situation may cause an important constraint, if, for example, the occurrence of the rightmost letter is delayed. Twenty years after McCreight, Ukkonen has presented an online version [27]. In the online construction algorithm, the scanned part of the sequence can be projected to the suffix tree whereas; it is possible to extend the suffix tree by reading the next letter from the sequence.

Advancement on the suffix tree construction took a step by Generalized Suffix Tree (GST) [3]. Bieganski looked at the problem from a different aspect and pointed out the importance of indexing multiple sequences in a single suffix tree. In GST, it was necessary to identify the origin of each sequence. Hence extra node identifiers were added within leaf nodes. As a result of GST, most of the symbolic representations of Time Series could be indexed by a single suffix tree [12].

Suffix tree construction on disk leads to important difficulties such as high space consumption and poor memory locality. Concretely, space consumption of a suffix tree node is high and fewer nodes can fit into a page. If a suffix tree contains large number of nodes, disk page requirement of a suffix tree will be large as well. On the other hand, poor memory locality is inevitable since suffix tree nodes are generated in random order and nodes of a selected path are generally spread across different pages. Because of this, traversal on a path frequently leads to indispensable page misses.

Recently, some researchers pointed out disk based suffix tree algorithms. In 1997, Farach-Colton proposed a theoretical algorithm, which ensured linear time construction, [7] but his algorithm has not supported by practical results. In PJama platform, authors suggested a space efficient algorithm by removing suffix links from suffix tree [13]. In addition they grouped the suffixes of a text according to their common prefixes. Therefore, suffixes in the same group can be inserted into the tree one by one. Hence both poor memory locality and space consumption drawbacks would be improved. Recently, new studies have followed a similar path [4, 22, 26, 29]. Nevertheless, these algorithms did not consider online property of suffix trees. and put constraints on dynamic sequence insertions. Although [23] introduces an online algorithm, it is not incremental and put constraints on streaming sequence insertions. In fact all these algorithms were designed for large genomic data sets and do not consider medium length streaming sequences such as MIDI.

In 2004, Bedathur and Haritsa presented an online suffix tree construction algorithm on disk [2] Based on Ukkonen's strategy, they considered physical node representations on a tree. In addition, they introduced a buffering strategy. Nonetheless, they did not test medium-size streaming sequences.

In this study, we present an Online Generalized Suffix Tree (OGST) approach on disk. To the best of our knowledge, this is the first study dealing with OGST construction on secondary memory. Briefly, contribution of this study is threefold: First, we modify the suffix node trees so that direct access to parent becomes possible. Second,

we introduce a sequence insertion strategy which is determined by semantic information of the text. Hence, we enable fast access to the frequently accessed nodes of the tree. Third, we show the relation between buffering performance and semantic information. In order to evaluate our approach, we make use of a popular MIDI database which contains four thousand musical sequences, where alphabet size is 128.

The remainder of the paper is organized as follows: section 2 introduces some basic definitions and explains alignment of online suffix tree construction on memory. In Section 3, we introduce a new physical node representation. Also, we analyze the contribution of alphabet letter frequencies. Consequently, we present an improved sequence insertion order strategy. Test results are demonstrated in Section 4. Finally, Section 5 concludes the paper and gives a look to further studies on this subject.

2 Online Generalized Suffix Tree Construction

This section provides an overview of Online Generalized Suffix Trees (OGST) and analyzes the factors affecting the performance of disk based suffix trees.

2.1 Definitions

Suffix trees enable fast string processing on large data sequences. As shown in Figure 1, suffix tree is composed of edges, internal nodes, and leaf nodes. In particular, edges connect the nodes of tree and represents subsequence of a suffix. Any unique path which connects the root and an internal node implies a common prefix. Particularly, a leaf node addresses a unique suffix from text.

There are several ways to construct suffix trees. One of them is online construction, where scanned part of the sequence can be immediately projected to the suffix tree. Besides, generalized suffix trees index multiple sequences.

In this subsection, some of the preliminary definitions about OGST are given to clarify the notation used throughout the paper. Let us assume that we have a collection of sequences, S, such that.

$$S = \{S^1, S^2, \ldots, S^k\} \tag{1}$$

Here, an arbitrary sequence, S^j, is defined as an ordered-set containing all possible suffixes defined in alphabet, Σ. More formally, an arbitrary sequence is defined as follows:

$$S = \left\{s_1^j, s_2^j, \ldots, s_n^j\right\} \tag{2}$$

where an arbitrary suffix, S_i^j, is a sequence containing the only last (n-i+1) letters of in the same order. Meanwhile, the alphabet of the data set containing letters, Σ, is defined as follows:

$$\Sigma = \{ \sigma_1, \sigma_2, \ldots, \sigma_\gamma \} \tag{3}$$

where the alphabet size is equal to γ, (i.e., $|\Sigma|=\gamma$).

2.2 Online Generalized Suffix Tree Construction on Disk

While suffix trees introduced salient enhancements on string processing, most of the major research in this area were concentrated on memory aspects [18][27]. A decade ago, it was widely believed that its disk based implementations were not feasible due to memory bottleneck [2]. Here, we analyze the reasons of memory bottleneck and later propose a new solution to enhance it.

The memory bottleneck is due to two factors. First one is high space consumption. Indeed, space cost of suffix trees range within 17n to 65n bytes [29]. For this reason, suffix trees can not fit into memory for large sequences. The second factor to cause memory bottleneck is the poor memory locality of nodes inside the tree. When a new common prefix occurs among suffixes, a representative internal node is generated and inserted into tree. Since generation order of the internal nodes are random and solely depends on the common prefix occurrences, random distribution of suffix tree nodes are indispensable and leads to poor memory locality.

Figure 1 illustrates a typical suffix tree. In Figure 1-a, suffixes of "effect" are indexed by a suffix tree. While dashed circles represent leaf nodes, flat circles denote internal nodes. The edges, which connect nodes, represent subsequences of a suffix. In order to illustrate poor memory locality, we illustrate node generation orders inside of circles. Therefore, we assume that initially root is generated and its generation order becomes 0. In the tree, nodes of the paths have irregular node generation order. Figure 1-b shows the tree after inserting a new suffix, "effort". Due to new common prefix between new suffix and the tree, a new internal will be generated. Since largest generation order in the tree was 9, generation order of the new node becomes 10.

Tree traversal is a technique for processing the connected nodes of a tree in some order [21]. For each suffix insertion, it is necessary to follow a top down tree traversal

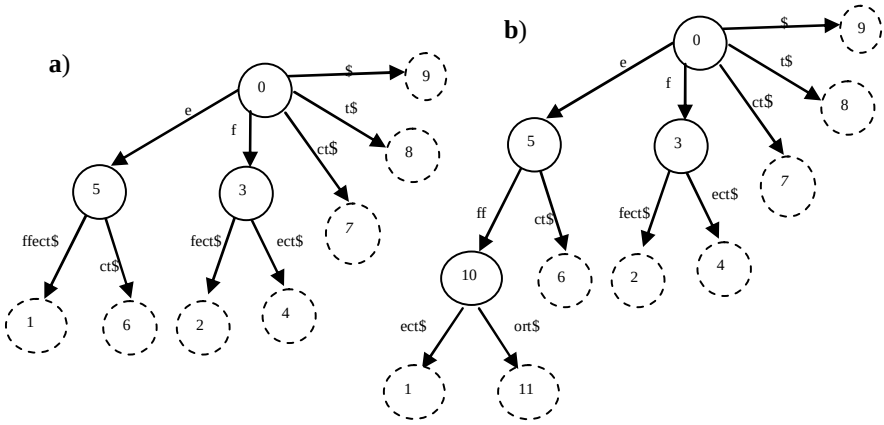


Fig. 1. a) Suffix tree after inserting all suffixes of the sequence "effect". Dashed circles denote leaf nodes; meanwhile flat line circles represent internal nodes. Inside the circles, node generation orders are pointed out. The delimiter character, $, is used to maintain the end of a sequence**. b)** The suffix tree after insertion of the suffix: "effort$".

path which starts from root. Nevertheless, if the depth of the tree is big, following a path before a node insertion takes O(n) time. Therefore, suffix tree construction is expensive. Instead, a shortcut from the previous node generation step can enable direct access to the demanded tree position. Here, corresponding shortcuts are named as suffix links. For a detailed description of linear time suffix tree construction and suffix links, reader is referred to read [11].

In an online suffix tree construction algorithm, physical node representations take more attention. An internal node may have so many children, or it may have only two children. Furthermore, total children of an internal node may increase during the suffix tree construction. We expect that physical node representation optimizes performance for possible cases.

2.3 Physical Representation of Suffix Tree Nodes

In a suffix tree, there are two fundamental and applicable choices for the physical representation of the suffix tree nodes: Linked-list Representation and Array Based Representation [2, 14]. The former strategy reduces space cost of an internal node; therefore it will be convenient for limited memory applications. However, from a disk based aspect, priorities change. Performance on disk depends solely on total pages misses [9]. The latter strategy enforces nodes to contain more information about children addresses; as a result, it ensures less number of hops to access to the next node in the path. The tradeoff between two strategies is an important factor for disk based application.

During the node generation, an internal node initially obtains two branches to handle child nodes. In other words, an internal node initially contains two child pointers and corresponding branches. While suffix tree construction proceeds, the node may obtain new children and consequent branches. For each new child, the node will need to keep an extra pointer. Henceforth, space requirement of an internal node increases. For instance, in Figure 1-b, the root node has five branches and consequent children nodes. Certainly, there is a limit on maximum branches from a node. If $|\Sigma|=5$, than an internal node will contain maximum five children.

In the suffix tree, alphabet size determines the maximum number of child pointers. The difficulty is holding possible $|\Sigma|$ branches in each internal node; at the same time, optimizing the ratio of used/unused child pointers.

Array Based Representation. This is a static data structure. By the time of an internal node generation, all possible $|\Sigma|$ child pointers will be arranged within an internal node; no matter how many of them are used. So, array-based representation simplifies the future modifications on the internal nodes. In Figure 2-a, alphabet has $|\Sigma|$ letters, rightmost $|\Sigma|$ pointers address possible branches. Start offset and edge length fields are aligned to represent subsequence of the sequence. In brief, each internal node of a generalized suffix tree requires $4+|\Sigma|$ pointers.

Linked List Based Node Representation. In contrast to array based representation, Linked-List representation does not hold the address of all possible children; but first child. Meanwhile siblings of a common parent are connected by links. Throughout this study, we name any linked list inside the suffix tree as “sibling list”. We denote a linked list node representation in figure 3, where only head of the sibling list is able to access to other sibling nodes

In terms of List Based Representation, handling parent address does not cause extra space consumption. Instead, we can handle the parent address with a sibling pointer since the tail node of a sibling list does not have a sibling.

3 Fast and Space Efficient ST Construction Algorithm on Disk

In a disk based OGST implementation, we believe that three issues are very important: (1) memory utilization, (2) fast access to a child node and (3) fast access to parent. Here, we present a technique which has two legs. In the first leg, we deal with direct access to a parent node. In the second leg, we sacrifice direct access to the child node for the sake of space utilization. However we can still present a strategy which enables fast access to a child node. To do this, we consider occurrence frequency of letters and probabilistic sequence occurrences.

3.1 Direct Access to Parent and Children Nodes

In order to optimize space utilization, we prefer a Linked List based node representation. However, we modify the nodes of Linked List by appending a parent address pointer and ensure direct access to parent node, named as Parent Address Appended List Representation (PAAL). As a result of this modification, size of an internal node will be increased by 20%. In Figure 2-c, we denote PAAL as an alternative physical node representation.

In contrast to direct access to parent, direct access to children is problematic. Although reserving a pointer for each possible child is possible; it may not be feasible due to additional space overhead. As aforementioned, reserving a child pointer for each letter is not feasible. Instead, we prefer space efficient linked list node representation where sibling nodes are connected. However, we aim to place the most frequently accessed nodes to the head side of linked lists. Therefore, we maintain a speed up on access time of frequently access nodes. On the other hand, we try to align rarely referenced nodes at the tail of sibling list and venture their expensive access cost.

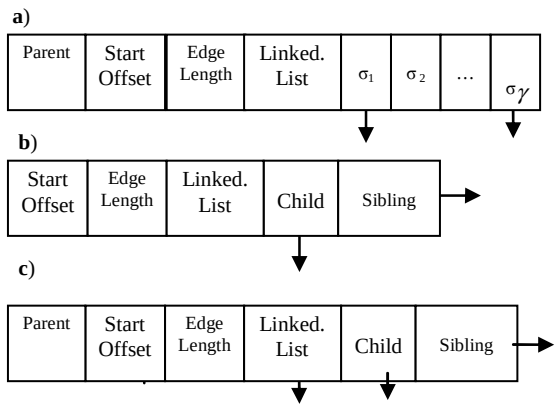


Fig. 2. Three different way of physical representation of an internal node: (a)Array based, (b)Linked List based, (c) PAAL

3.2 Impact of Alphabet Size on Tree Construction

Alphabet size may have deep impact over the suffix tree construction performance. In a static array node representation, large alphabets increase the size of internal nodes whereas; in a linked list representation they lead to an increase on the length of sibling lists. We need to mention that most of the studies from literature aim to index DNA sequences where alphabet size is only four. Therefore, space utilization of array based representation leads to acceptable ratios. Nevertheless, conditions are different when alphabet size is large. For instance MIDI alphabet contains 128 letters.

If we assume that nodes of a common parent are stored in the same disk page, Linked List based representation outperforms. Nonetheless poor memory locality of nodes reduces performance on disk. As in Figure 3, access to $v_{i\gamma}$ can be very expensive when alphabet size is large, since each node access may cause a page miss to disk.

3.3 Impact of Letter Frequency Distribution on Tree Traversal

Probabilistic occurrences of letters in the alphabets are generally different. The English language is a good example. In English, 'E' is the most frequently used letter. In contrast, 'Q' is the letter whose occurrence frequency is the least. In 1830's, Morse alphabet is inspired by such information. Similarly learning such information from a domain expert can enhance suffix tree construction for music sequences.

In PAAL node representation, child access cost depends on traversals on sibling lists. In the Figure 3, the node which is the head of the sibling list, vi1, can be accessed by one hop. However, accessing the node v_{iN}, which is the tail of sibling list causes extra traversal costs. It is preferred to see that v_{i1} is the most frequently referenced node, whereas access to the v_{iN} is very rare.

In the suffix tree, a less frequently used letter constitutes comparably simpler branches. As a result, depths of the relevant branches are comparably shallow and short. In contrast, branches those starts with dominant letters are complex and deep. Therefore, probability of reading such braches in the future is comparably higher

3.4 Probabilistic Occurrence of Longest Common Prefix

In this section, we explain which nodes of the suffix tree are accessed frequently. To do this we consider letter occurrence probabilities. Namely, we set about occurrence probability of all possible common prefixes. Therefore frequently accessed nodes can be estimated while suffix tree construction proceeds.

Lemma 1: For a given depth of a node is, p, the sequence between root and this node has at least p letters.

Proof: Since depth of a node is p, there exist exactly p edges to connect nodes on the same path. As it mentioned before, edges contain at least one letter. Hence the sequence between root and this node has at least *p* letters.

Lemma 1 implies that depth of a node in the tree leads decent impact over the sibling list length. Nodes in the higher end side of the tree commonly have more siblings and length of the sibling list in the higher end side of the tree is long. Therefore access to the tail of a sibling list more number of node access. On the other hand, sibling list

are shorter in deeper side no matter the length of the alphabet. For this reason, the probability of a common prefix comes into prominence in a tree level and length of the corresponding sibling list comes to the prominence.

3.5 Alignment of Sibling Nodes to Enhance Memory Locality

In order to reduce side effects of poor memory locality, we try to postpone constructing the specific branches of the tree and delay the insertion of relevant sequences. In order to illustrate this, we present a primitive alphabet and make the following assumptions Let there exist two alphabets, Σ_1 and Σ_2 satisfying the following properties:

$\Sigma_1 \cup \Sigma_2 = \Sigma$, $\Sigma_1 \cup \Sigma_2 = \emptyset$ and $|\Sigma_1| = |\Sigma_2| = |\Sigma|/2$

We also assume that there exist three sequences, S_1, S_2, and S_3. All letters of S_1 and S_2 comes from Σ_1 and Σ_2, respectively. On the other hand, S_3 contains letters from Σ. In order to introduce the effect of data set size, we assume that length of S_1 is longer than the length of S_2. All three sets are planned be inserted to the same suffix tree.

Lemma 2: Maximum tree construction performance will be obtained, if we insert sequences of sets in the following order: First insert S_1, later S_2, and finally S_3.

Proof: Cost of an unsuccessful node search depends on the length. The shorter the sibling list, the faster the search time. In the first phase, we insert S_1, where length of siblings list cannot exceed $|\Sigma|/2$. Hence inserting the S_1 can be done quickly.

In the second phase we insert the set S_2 into tree and maximum length of the sibling list will be extended to $|\Sigma|$. Still, cost of a successful search is $O(|\Sigma|/2)$, since nodes which are generated in the first phase takes place in tail side of sibling list and successful searches never visit them. However, cost of an unsuccessful search time increases to $O(|\Sigma|)$. Because of this fact large sequence set, S_1, should be inserted to the tree before S_2.

At last, when we insert S_3, we will encounter longest sibling list. Prior, most frequently referenced sibling lists should have already contained $|\Sigma|$ elements. Hence, both successful and unsuccessful search time will be proportional to $|\Sigma|$.

3.6 Computing the Rank of a Sequence and Inserting to the Suffix Tree

When a new node is generated, it will be appended to the relevant sibling list as the new head. Correspondingly, rarely accessed nodes should be generated earlier to takes place in tail side of sibling lists. In this section, we ensure this by introducing a sequence insertion order strategy.

As in the English letters, we assume that average occurrence frequency of each letter and its corresponding histogram is known. This can be learned from a domain expert. In midi music databases, highest and deepest pitches have lower occurrence probability than the rest. Hence we collect sequences which densely contain such pitches at S_1. Therefore rarely accessed nodes of the tree will be generated first and took place at the tail side of sibling lists. On the other hand, generation of the frequently accessed nodes of the tree will be delayed. Consequently frequently accessed nodes can stand as the head of the sibling lists of tree after inserting a group of sequences into tree.

We need to emphasize that each internal node becomes head of a sibling list just after its generation. We prefer that all accesses to the new node should be processed before another node is appended to the same sibling. In other words, we should try to construct all sections of a branch before locating to another branch.

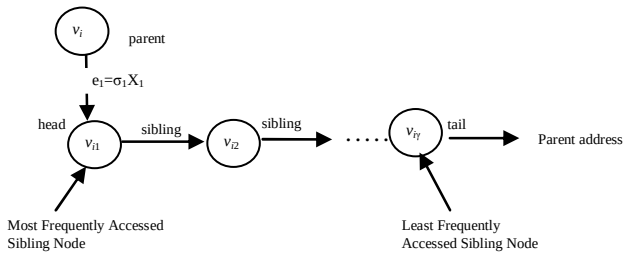


Fig. 3. Sibling list for a common parent. While direct access to the first child is possible, access to requires traversal on sibling list.

In the future, dynamic sequence insertions may continue and lead to generation of new nodes in random order. However, it will not cause a big problem since nodes in higher end side of the tree will have already been already organized.

4 Experimental Results

In this section, we present the results experimentations, which evaluate the physical node representation techniques on a disk based suffix tree. Besides, we consider the effect of buffering techniques and page size. Our evaluation criterion is based on page hits and misses. In our experiments, cost of a pointer is four bytes. Since space consumption of internal nodes and leaf nodes are different, we distinguish internal and leaf nodes in disk. In other words, a page is composed either all leaf nodes or internal nodes. Unless mentioned, we assume that size of a page is 4096 bytes. However, compare the performances of different page sizes.

In order to test the approaches, we make use of Digital Tradition Folk Music Database, containing nearly 4000 music files, whereas standard MIDI alphabet contains 128 letters.

4.1 Comparison of Physical Node Representation Approaches

In Figure 4-a, we denote the space consumption of various node representation techniques. In the figure, Linked List representation ensures the best space utilization. Meanwhile, PAAL representation yields satisfactory space utilization as well. However, space consumption of the Static Array Representation is quite high. Basic factor of the worst space utilization is the overheads in internal nodes. As aforementioned, an internal node needs to maintain $|\Sigma|+4$ pointers. In terms of MIDI sequences, high space consumption is indispensable since alphabet size is quite large (i.e., $|\Sigma| = 128$). Due to this fact, performance of Static Array is even worse if the alphabet size increases.

In terms of suffix tree construction speed, total page miss occurrence comes into prominence. Concretely, the fewer the page misses, the faster the approach is. In Figure 4-b, we denote the total page misses caused by physical node representation techniques when buffering is not considered. Although Linked List node representation is space efficient, it leads to high amount of page misses; hence it cannot be feasible on disk. Basic factor behind the performance loss is the node traversals on sibling lists. On the other hand, Static Array outperforms and ensures to least number of page misses. In fact, Static Array Representation enables direct access to the child or parent node. Meanwhile our PAAL introduces compromise between Static Array and Linked List.

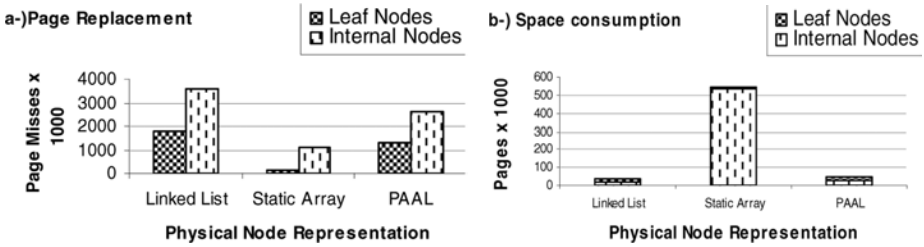


Fig. 4. Physical node representation versus a) Page Replacement and b) Space Consumption

4.2 Effect of Buffering

In terms of suffix tree indexing, page requirement of a large file is enormous. Consequently, data will be aligned into millions of pages. Due to random distribution of the nodes; probability of finding two consecutive nodes in the same page is very small. The chance can be increased by buffering. In this section, we evaluate the effect of buffering on both vertical and horizontal traversals and discuss the contribution of letter frequency based sequence insertions on buffering.

Here, we compare the performance of three physical node representations using the following page replacement policies:

- Least Recently Used (LRU): If a page fault encounters, least recently used page will be replaced.
- First In First Out (FIFO): In case a page fault occurs, the page which stayed longest in buffer is replaced.
- TOP_Q : Replaces the page if the average depth of the nodes in the page is the highest. In addition, replaced nodes will not be dropped immediately; instead they are processed in a FIFO fashioned buffer. We assume that 20 % of the buffer is reserved for FIFO buffer.

In Figure 5, we compare three physical node representations approaches and obtain results when the buffer contains 16, 64, 256 and 1024 pages. Test results imply that Linked List Representation yields the worst page miss ratio. In contrast, Static Array Representation outperforms and ensures least page misses when buffer contain 16 pages. However, increasing the total pages in buffer does not enhance its performance. Instead,

increasing size of the buffer drastically rehabilitates the performance of PAAL. All in Figures 5-a,b,c we observe that PAAL ensures least number of page misses when buffer contains 1024 pages, hence outperforms. From the results, we can conclude that buffering cannot expose its positive effect if the disk space is used extravagantly as in Static Array node representation.

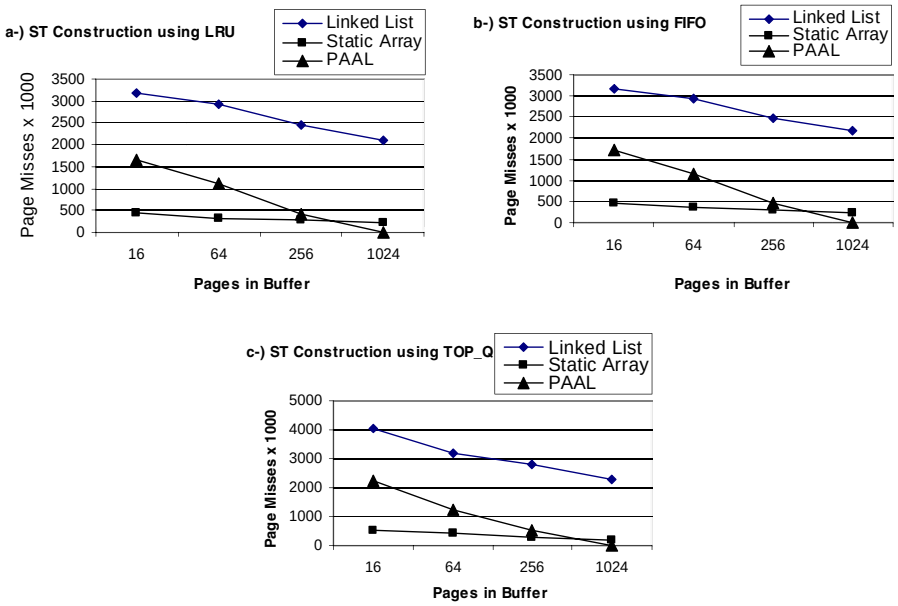


Fig. 5. Comparison of the node representation algorithms in terms of page misses*1000. a) LRU buffering b) FIFO buffering c) TOP_Q buffering.

4.3 Effect of the Page Size

Currently, disk based suffix trees are mostly tested on 4K pages. In this section, we analyze the contribution of page size over construction speed. For a constant buffer space, we compared the performance of variable page sizes. In Figure 6, we evaluate performance of 2K, 4K, 8K and 16K and reveal their page miss occurrences. Here, our performance criteria is data transfer time (dtt) since page miss cost of 2K and 16 K is not same.

Figure 6-a, illustrates the performance under the condition that physical node representation is based on PAAL. In the figure, we see that large page size reduce total page misses. However, data transfer time of a 16K page is eight times more than a 2K page. Hence small page size outperforms. Similarly, Figure 6-b and 6-c support the same result as well. Besides, Figure 6 illustrates that buffer management strategies yield various outcomes if underlying physical node representation changes. For instance, Figure 6-c denotes that Static Array Representation prefers TOP_Q buffering strategy. In contrast, TOP_Q is not convenient for Linked List Representation.

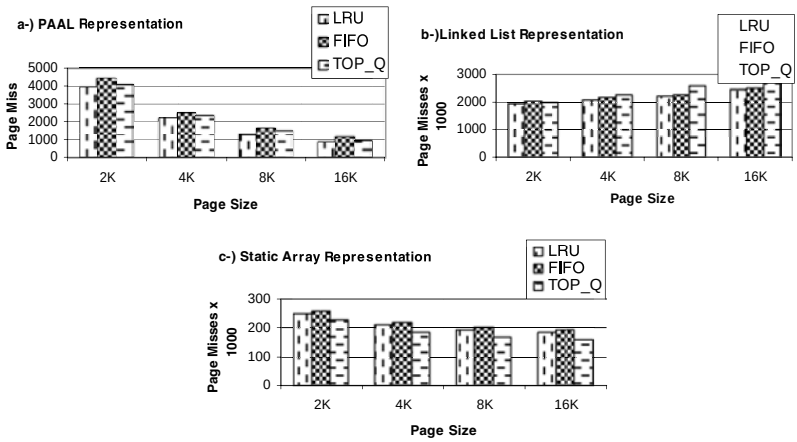


Fig. 6. Impact of page size over page miss occurrences where underlying physical node representation is a) PAAL, b) Linked List c) Static Array

5 Conclusion

In this study, we proposed a novel approach for Online Generalized Suffix Tree (OGST) construction on secondary memory. We showed that poor memory locality is indispensable when online generalized suffix tree construction is implemented. Moreover, we showed that large alphabet size drastically drops the performance on secondary memory. To solve this problem, we proposed a space efficient physical node representation, named as PAAL, to enable direct access to the parent. In order to speed up child access time, our approach takes access frequency of children into consideration Therefore; children of a common parent are aligned on a sibling list depending on their estimated access frequencies. In this study, we estimate the access probability of child nodes by letter occurrence frequencies of the alphabet. In contrast to expectations, we assign higher insertion priority to the sequences those contain least frequently used letters of the alphabet. Consequently, least frequently used children of a common parent are aligned in a position approaching to the tail of sibling lists. In this way, new sequence insertions to the suffix tree yield better performance.

References

[1] Abouelhoda, M.I., Kurtz, S., Ohlebusch, E.: Replacing suffix trees with enhanced suffix arrays. Journal of Discrete Algorithms 2 (2004)
[2] Bedathur, S., Haritsa, J.: Engineering a fast online persistent suffix tree construction. In: Proceedings of ICDE (2004)
[3] Bieganski, J.R.P., Carlis, J.V.: Generalized suffix trees for biological sequence data: Application and implantation. In: Proc. of 27th HICSS. IEEE, Hawai (1994)
[4] Cheung, C.-F., Yu, J.X., Lu, H.: Constructing suffix tree for gigabyte sequences with megabyte memory. IEEE Transactions on Knowledge and Data Engineering (2005)

[5] Clifford, R., Sergot, M.J.: Distributed and paged suffix trees for large genetic databases. In: Baeza-Yates, R., Chávez, E., Crochemore, M. (eds.) CPM 2003. LNCS, vol. 2676. Springer, Heidelberg (2003)
[6] Cormen, T.H., Leiserson, C.E., Rivest, R.L.: Introduction to Algorithms. The MIT Press, Boston (1989)
[7] Farach, M., Ferragina, P., Muthukrishnan, S.: Overcoming the memory bottleneck in suffix tree construction. In: 39th Symp. on Foundations of Computer Science. IEEE Computer Society, Los Alamitos (1998)
[8] Ferragina, P., Grossi, R., Montangero, M.: A note on updating suffix tree labels. Theoretical Computer Science (1998)
[9] Folk, M., Riccardi, G., Zoellick, B.: File structures: an object-oriented approach with C++, 3rd edn. Addison-Wesley Longman Publishing, Amsterdam (1997)
[10] Giegerich, R., Kurtz, S.: From Ukkonen to McCreight and Weiner: a unifying view of linear-time suffix tree construction. Algorithmica 19(3), 331–353 (1997)
[11] Gusfield, D.: Algorithms on strings, trees, and sequences Computer Science and Computational Biology. Cambridge Univ. Press, Cambridge (1997)
[12] Huang, Y.-W., Yu, P.S.: Adaptive query processing for time-series data. In: Proceedings of KDD. ACM Press, New York (1999)
[13] Hunt, E., Atkinson, M.P., Irving, R.W.: A database index to large biological sequences. In: 27th Int'l Conf. Very Large Data Bases. ACM Press, New York (2001)
[14] Kurtz, S.: Reducing the space requirement of suffix trees. Software—Practice & Experience 29(13), 1149–1171 (1999)
[15] Lemström, K.: String matching techniques for music retrieval, PhD thesis, University of Helsinki, Department of Computer Science (November 2000)
[16] Manber, U., Myers, G.: Suffix arrays: a new method for on-line string searches. SIAM Journal on Computing (1993)
[17] Martinez, H.M.: An efficient method for indexing repeats in molecular sequences. Nucleic Acids Research (1983)
[18] McCreight, E.M.: A Space-economical suffix tree construction algorithm. Journal of ACM 23 (1976)
[19] Munro, J.I., Raman, V., Rao, S.: Space efficient suffix trees. J. of Algorithms 2 (2001)
[20] Navarro, G.: A guided tour to approximate string matching. ACM Computing Surveys
[21] `http://www.nist.gov/dads/HTML/treetravrsl.html`
[22] Phoophakdee, B., Zaki, M.: Genome-scale disk based suffix tree indexing. In: Proceedings of ACM SIGMOD (2007)
[23] Sandeep, A., Akinapelli, S.: Online construction of search-friendly persistent suffix-tree layouts. M.Sc thesis, Indian Institute of Science Bangalore (July 2006)
[24] Salzberg, B.: File Structures: An analytic approach. Prentice-Hall, Englewood Cliffs (1988)
[25] Schürmann, K., Stoye, J.: Suffix tree construction and storage with limited main memory. unpublished technical report, Univ. Biefeld (2003)
[26] Tian, Y., Tata, S., Hankins, R.A., Patel, J.M.: Practical methods for constructing suffix trees. The VLDB Journal (2005)
[27] Ukkonen, E.: On-line construction of suffix-trees. Algorithmica (1995)
[28] Weiner, P.: Linear pattern matching algorithm. In: Proc. of 14th IEEE Symp. On Switching and Automata Theory (1973)
[29] Wong, S., Sung, W., Wong, L.: CPS-tree: A compact partitioned suffix tree for disk based indexing on large genome sequences. In: Proc. of IEEE ICDE, Istanbul (2007)

Fuzzy Voronoi Diagram

Mohammadreza Jooyandeh and Ali Mohades Khorasani

Mathematics and Computer Science,
Amirkabir University of Technology,
Hafez Ave., Tehran, Iran
mohammadreza@jooyandeh.info, mohades@aut.ac.ir
http://math-cs.aut.ac.ir

Abstract. In this paper, with first introduce a new extension of Voronoi diagram. We assume Voronoi sites to be fuzzy sets and then define Voronoi diagram for this kind of sites, and provide an algorithm for computing this diagram for fuzzy sites. In the next part of the paper we change sites from set of points to set of fuzzy circles. Then we define the *fuzzy Voronoi diagram* for such sets and introduce an algorithm for computing it.

Keywords: Fuzzy Voronoi Diagram, Voronoi Diagram, Fuzzy Voronoi Cell, Fuzzy Geometry, Fuzzy Set.

1 Introduction

Fuzzy objects becomes to be focused after 1965 when Zadeh introduced Fuzzy set for the first time and after that it becomes a part of other fields. In this paper we work on Fuzzy Voronoi diagrams. It's an important task for two reasons. First that this diagram will be helpful in Fuzzy spaces, and somehow in Probabilistic spaces. Second is that Voronoi diagram is used in other fields and even other sciences. So defining this kind of diagram will solve the same problems when other fields switch context to fuzzy ones. Voronoi diagram is studied in some extensions. These extensions are based on changing the meter of the space, dimension of the space or sites of the diagram [1], [2], [3]. Also some other Voronoi diagrams are introduced, like weighted Voronoi diagram in [4] and approximate version of Voronoi diagram in [5]. In this paper we first change the set of sites to fuzzy set and then to set of fuzzy circles, and introduce Voronoi diagrams for these types of sets and provider algorithms for computing them.

Definition 1. *Let P be a discrete subset of a metric space like X. For every point p in P, the set of all points x in X which their distance from p is lower (or equal) to other points of P is said to its Voronoi cell (or Diricle domain) and be shown by $V(p)$. In mathematical words:*

$$V(p) = \{x \in X \,|\forall q \in P\, [d(x,p) \leq d(x,q)]\,\} \tag{1}$$

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 82–89, 2008.

Definition 2. *Let P be a discrete subset of a metric space like X. Voronoi diagram of P will be the set of all Voronoi cells of its points, which is shown by $V(P)$. Members of P also called Voronoi cite. In mathematical words:*

$$V(P) = \{V(p) \,|p \in P\} \tag{2}$$

Because of topological properties of this set we can equivalently define the Voronoi diagram as set of boundary of it. In this paper we work on the second definition. Also this boundary could be the points which have the property that, they have maximum distance from the nearest site(s). If we assume $\mathbb{R}^2$ as the space and use Euclidian meter, this boundary would be some line segments. An example of such diagrams is shown in Figure 1.

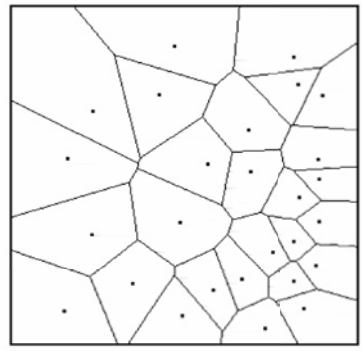


Fig. 1. A Sample Voronoi Diagram for a Set of Points

2 Fuzzy Voronoi Diagram

Definition 3. *Let $\tilde{P}$ be any fuzzy set. The non-fuzzy image of it will be shown by P (with the same Alphabet, without tilde), and will be defined as follows:*

$$P = \left\{ x \,\middle|\, \exists \lambda \; (x, \lambda) \in \tilde{P} \right\} \tag{3}$$

Definition 4. *Let to be a fuzzy subset of R2. Fuzzy Voronoi cell of a point will be defined as follows:*

$$\tilde{V}(\tilde{P}) = \left\{ (x, \chi_P(x)) \,\middle|\, \forall \tilde{q} \in \tilde{P} \, [d(x,p) \leq d(x,q)] \right\} \tag{4}$$

Definition 5. *Let $\tilde{P}$ be a fuzzy subset of $\mathbb{R}^2$. Fuzzy Voronoi diagram of it will be defined as follows. Let (x, α) be a member of boundary of fuzzy Voronoi cell of a point p in P, (x, φ) will be a member of fuzzy Voronoi diagram of $\tilde{P}$ in which:*

$$\varphi = \underset{(x,\alpha) \in \tilde{V}(\tilde{p}) \wedge \tilde{p} \in \tilde{P}}{\top} (\alpha) \tag{5}$$

Note: In this paper we use $\top$ and $\perp$ for T-Norm and S-Norm.

Theorem 1. *Non-fuzzy image of fuzzy Voronoi diagram of a fuzzy set of sites is Voronoi diagram of non-fuzzy image of the set of sites.*

Proof. Let (x, α) be a member of fuzzy Voronoi diagram of $\tilde{P}$. So this point is member of at least two fuzzy Voronoi cell like Voronoi cells of $\tilde{p_{i_1}}$ and $\tilde{p_{i_2}}$. Based on Definition 3, $d(x, p_{i_1}) = d(x, p_{i_2})$ and also

$$\forall \tilde{q} \in \tilde{P} \left[d(x, p_{i_1}) \leq d(x, q) \right] \tag{6}$$

So we have simliar result for P:

$$\forall q \in P \left[d(x, p_{i_1}) \leq d(x, q) \right] \tag{7}$$

And because of Definition 2, x would be a member of $V(P)$.

Reverse assume that x be a member of $V(P)$. So there must exists sites $p_{i_1}, \ldots, p_{i_n}$ $(n \geq 2)$ such that x be a member of their Voronoi cells, and without loss of generality assume $p_{i_1}, \ldots, p_{i_n}$ be all of such sites. So for any j in the range, $\left(\chi_P\left(p_{i_j}\right)\right)$ would be a member of $\tilde{V}(\tilde{p_{i_j}})$. So based one Definition 5:

$$\left(x, \underset{1 \leq j \leq n}{\top} \left(\chi_P\left(p_{i_j}\right)\right) \right) \in \tilde{V}(\tilde{P}) \tag{8}$$

And also $\chi_P\left(p_{i_j}\right) > 0$ holds for all j which garanties:

$$\underset{1 \leq j \leq n}{\top} \left(\chi_P\left(p_{i_j}\right)\right) > 0 \tag{9}$$

3 Algorithm for Fuzzy Voronoi Diagram

According to Theorem 1 we can compute Fuzzy Voronoi diagram of a fuzzy set easily. We can compute the classic Voronoi diagram for the non-Fuzzy image of set of sites according to Fortune algorithm [6] and while computing the diagram we can compute the degree of membership for each line segment in $O(1)$ because any line segment will be boundary of two Voronoi cell so computing of the $\top$ will cost $O(1)$ time. So the total time of algorithm would be $O(n.\log(n)$ which the time of Fortune algorithm is.

4 Fuzzy Voronoi Diagram for Fuzzy Circles

In the previous section we introduce an algorithm for computing Fuzzy Voronoi diagram for Fuzzy set of points. In this section we extend the definition of Voronoi diagram to cover a larger set of objects not only points. Assume that the members of the set of sites become itself sets. It means that each site is a set of points which can be any geometric object like circle, rectangle and etc. This extension of Voronoi diagram for no-Fuzzy objects is studied in many researches like [4], [5], [7]. But the Voronoi diagram of a set of Fuzzy objects is not much similar to non-Fuzzy version like previous sections. To clarify the difference, consider the

following example. Assume that we have only two sites which both are circles with equal radius which both have continues characteristic function. And assume that the degree of membership converges to zero in boundary of circles. Now let we want to have non-fuzzy Voronoi diagram for non-fuzzy image of these sites, it's clear that the diagram would be bisector of their centers. But if we want to convert it to fuzzy one, like previous section; we see that the degree of membership of this line in the diagram would be zero because it made from points which membership degree is lower than any (according to the continuity of the membership functions which was assumed) so the line wouldn't be part of the diagram!

The problem occurs because the Voronoi diagram of fuzzy objects should be created using all of its points and according to their degree of membership. So we would have a diagram like Figure 2 for such sites.

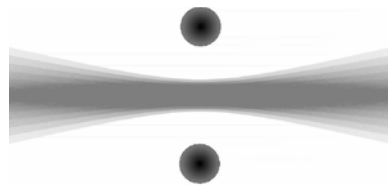


Fig. 2. Fuzzy Voronoi Diagram of two Fuzzy Circle

Definition 6. *Let $\tilde{P}$ be a finite family of fuzzy subsets of $\mathbb{R}^2$, then $\tilde{x}$ would be a member of its fuzzy Voronoi diagram $\tilde{V}(\tilde{P})$, iff there exists $\tilde{p_i}$ and $\tilde{p_j}$ in $\tilde{P}$ such that the distances of x from at least one pair of points like x_i and x_j in them be equal and also $\tilde{p_i}$ and $\tilde{p_j}$ be two of the most nearest sites to x. The degree of membership of x in $\tilde{V}(\tilde{P})$ would be $\perp$ of all such pairs.*

Definition 7. *A fuzzy subset of $\mathbb{R}^2$ is called a fuzzy circle iff there exists a disk in $\mathbb{R}^2$ such that:*

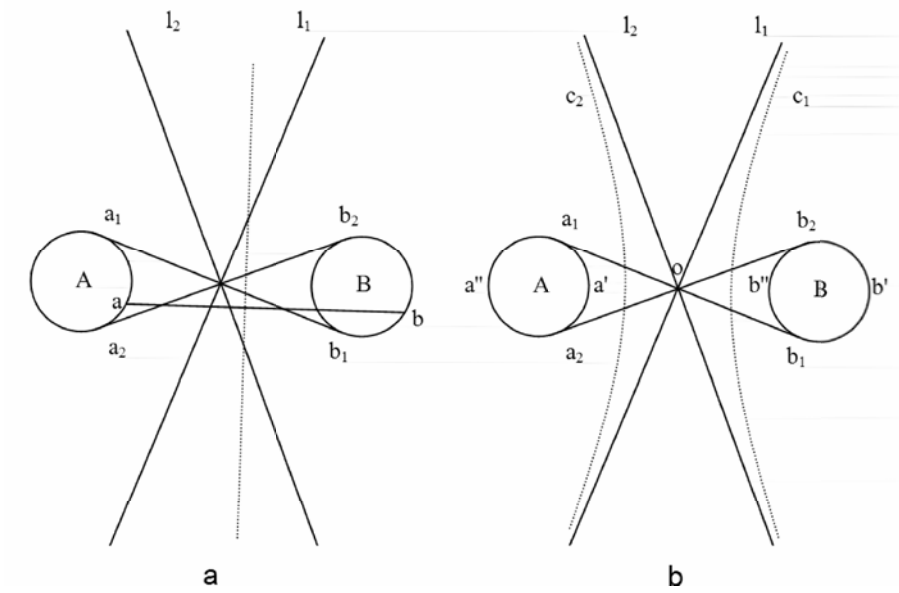
- *All points of the set are member of the disk.*
- *In any open neighborhood of a boundary point of the disk, there exists at least one point of the fuzzy set (with membership degree greater than zero).*

Note: If we remove the second condition from the above definition any bounded subset of $\mathbb{R}^2$ becomes a fuzzy circle.

It is the time to compute the Voronoi diagram for fuzzy circles. In this paper we assume all circles have the same radius and this section we assume that they boundary points have non zero degree of membership which can be simply proved that is not important because of the second condition of the Definition 7 and also connectedness of $\mathbb{R}^2$.

As the Definition 6 says the Voronoi diagram of two set will be fuzzy union of Voronoi diagram of pair of points which each one belongs to one the sets. But this sentence doesn't lead to an algorithm because this operation is not a computable operation. So we should analyze those diagrams to provide an algorithm.

In this part we introduce a method for computing the boundary of fuzzy Voronoi diagram of two fuzzy circles. We assume that this boundary will be created by boundary points of the circle and by Theorem 2 we prove this assumption. Let A and B be two fuzzy circle (Fig. 4.a and let a and b be a pair of point on their boundary. Slope of their connector line is clearly between slope of $\overline{a_1b_1}$ and $\overline{a_2b_2}$ which common tangents of these circles are. So slope of their bisectors would be between l_1's and l_2's. So these bisectors in a bounded interval is out of the area which is between l_1 and l_2.



Also the function which maps pair of points of A and B to their bisector is differentionable so that's enough to consider the boundary points, the farthest lines from o, which are made by (a', b') and (a'', b'') (Figure 4.b). These pairs make two lines which are perpendicular on connector line of center of circles and their distance is equal to diameter of the circles. And according to the result of above paragraph (bisectors are inside the area between l_1 and l_2 except in a bounded interval) and also continuity of the mapping function of pairs of points to the lines, we can assume a shape like dotted one in the Figure 4.b.

Now we should compute the boundary of the diagram not by introducing its shape! Let (A_x, A_y) and (B_x, B_y) be centers of circles and both radius be r. First we must compute points a_1, a_2, b_1 and b_2 which is simply done in $O(1)$. Then we must iterate arcs $a_1a'a_2$ and $b_1b'b_2$ simultaneously and compute their bisector lines. We use parametrization for simpler computation. For simplifying of the result let both circles be on a horizontal line. As it shown in Figure 4 while the angle changes on A from m to $2\pi - m$, the angle on B is changing from $\pi + m$ to $\pi - m$ and so:

$$a_x = A_x + r.\cos(t_1) \tag{10}$$

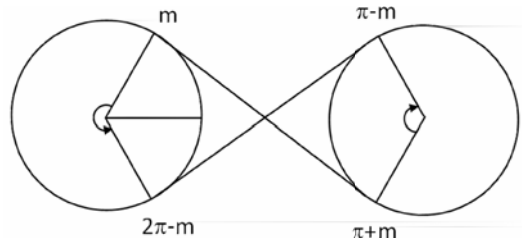


Fig. 3. Arcs on Adjacent Circles

$$a_y = A_y + r.\sin(t_1) \tag{11}$$
$$b_x = B_x + r.\cos(t_2) \tag{12}$$
$$b_y = B_y + r.\sin(t_2) \tag{13}$$

in which $t_2 = \frac{m}{m-\pi}(t_1 - m) + \pi + m$ and $m \leq t_1 \leq 2\pi - m$.

Now we should compute the points which each line makes on the boundary of the Voronoi diagram. For this purpose we find the intersection two lines which are infinitely near to each other. The next Maple program will compute the point for two neighbor circle.

```
ax1:=Ax+r*cos(t);
ay1:=Ay+r*sin(t);
ax2:=Ax+r*cos(t+e);
ay2:=Ay+r*sin(t+e);
bx1:=Bx+r*cos(m*(t-m)/(m-pi)+pi+m);
by1:=By+r*sin(m*(t-m)/(m-pi)+pi+m);
bx2:=Bx-r*cos(m*(t-m)/(m-pi)+pi+m+e);
by2:=By+r*sin(m*(t-m)/(m-pi)+pi+m+e);
y1:=(ay1+by1)/2+((ax1+bx1)/2-x)*((ax1-bx1)/(ay1-by1));
y2:=(ay2+by2)/2+((ax2+bx2)/2-x)*((ax2-bx2)/(ay2-by2));
x1:=(ax1+bx1)/2+((ay1+by1)/2-y)*((ay1-by1)/(ax1-bx1));
x2:=(ax2+bx2)/2+((ay2+by2)/2-y)*((ay2-by2)/(ax2-bx2));
xte:=solve(y1=y2,x);
yte:=solve(x1=x2,y);
xt:=limit(xte,e=0);
yt:=limit(yte,e=0);
simplify(yt);
simplify(xt);
```

Theorem 2. *The boundary of Voronoi diagram for empty circles and non empty ones are equal.*

Proof. Suppose a and b be a pair of points on circles A and B, and suppose one of them be inside of the circle. Without loss of generality assume that a be the interior point. Assume the connector line of a and b. This line will makes two points on the boundary of A. Assume that these points be a' and a''. Clearly

bisector of $\overline{ab}$ will be between bisectors of $\overline{a'b}$ and $\overline{a''b}$. So it completely places inside the Voronoi diagram area and it won't make a point on the boundary of the diagram.

5 Algorithm for Computing Fuzzy Voronoi Diagram for Fuzzy Circles

That's enough to compute the classic Voronoi diagram for the center of circles and then for each line segment in the diagram compute the fuzzy Voronoi diagram as described above and compute the intersection between two neighbors (which costs $O(1)$ for each pair). The correctness of the algorithm follows from Definition 6 in which only the most nearest sites has role in the diagram. For computing the intersection of two neighbors that's enough to compute the above Maple program twice (for example for circle A, B and then for A, C). Let their result be two pair (x1t1,y1t1) and (x2t2, y2t2) and their parameter be t1 and t2 then the following Maple command will compute the value of t1 and t2 in which the intersection happens. So can compute the intersection points from them.

```
solve({x2t2=x1t1,y2t2=y1t1},{t1,t2});
```

6 Conclusion

In this paper we introduce fuzzy Voronoi diagram and studied it for point and circle cells and showed that the diagram can be computed in $O(n\log(n))$ time for both and this will be helpful because of its application in other fields.

Some other works still can be done on this diagram and related topics which some of them are:

- Computing the degree of membership of interior points for fuzzy Voronoi diagram of circles
- Studying the dual of this diagram which may lead to extension of Delony Triangulation
- Studying the fuzzy Voronoi diagram for other shapes
- Studying the fuzzy Voronoi diagram with other meters or in higher dimensions
- [8] creates Voronoi diagram for finding the nearest point for a fuzzy controller maybe using fuzzy Voronoi diagram makes some improvements.

References

1. Leibon, G., Letscher, D.: Delaunay triangulations and Voronoi diagrams for Riemannian manifolds. In: 16th Annual Symp. Foundations on Computational Geometry, pp. 341–349. ACM Press, Hong Kong (2000)
2. Alt, H., Cheong, O., Vigneron, A.: The Voronoi diagram of curved objects. Discrete Comput. Geom. 34, 439–453 (2005)

3. Gavriloa, M.L., Rokne, J.: Updating the topology of the dynamic Voronoi diagram for spheres in Euclidean d-dimensional space. Discrete Compu. Geom. 20, 231–242 (2003)
4. Aurenhammer, F., Edelsbrunner, H.: An optimal algorithm for constructing the weighted Voronoi diagram in the plane. Pattern Recognition 17, 251–257 (1984)
5. Arya, S., Malamatos, T.: Linear-size approximate voronoi diagrams. In: 13th Annual ACM-SIAM Symp. on Discrete algorithms, pp. 147–155. Society for Industrial and Applied Mathematics, San Francisco (2002)
6. Fortune, S.: A sweepline algorithm for Voronoi diagrams. Algorithmica 2, 153–174 (1987)
7. Kim, D.S., Kim, D., Sugihara, K.: Voronoi diagram of a circle set from Voronoi diagram of a point set. Comput. Aided Geom. Des. 18, 563–585 (2001)
8. Kavka, C., Schoenauer, M.: Evolution of Voronoi-Based Fuzzy Controllers. In: Yao, X., Burke, E.K., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J.E., Tiňo, P., Kabán, A., Schwefel, H.-P. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 541–550. Springer, Heidelberg (2004)

A Novel Partitioned Encoding Scheme for Reducing Total Power Consumption of Parallel Bus

Mehdi Kamal, Somayyeh Koohi, and Shaahin Hessabi

Department of Computer Engineering, Sharif University of Technology,
Tehran, Iran
{kamal,koohi}@ce.sharif.edu, hessabi@sharif.edu

Abstract. Two main sources for power dissipation in parallel buses are data transitions on each wire and coupling between adjacent wires. There are many techniques for reducing the transition and coupling powers. These methods utilize extra control bits to manage the behavior of data transitions on parallel bus. In this paper, we propose a new coding scheme which tries to reduce power dissipation of control bits. The proposed method employs partitioned Bus Invert and Odd Even Bus Invert coding techniques. This method benefits from Particle Swarm Optimization (PSO) algorithm to efficiently partition the bus. In order to reduce transition and coupling power of control bits, it finds partitions with similar transition behaviors and groups them together. One extra control bit is added to each group of partitions. By properly managing transitions on control bits of each partition and that of each group, it reduces total power consumption, including coupling power. It also locates control bits of each partition such that total power consumption is minimized. We evaluate our method on both data and address buses. Experimental results show 40% power saving in coded data compared to original data. We also show the prominence of the proposed coding scheme over other techniques.

Keywords: Power, Coding, Parallel Bus, Coupling, Partitioning, PSO.

1 Introduction

There are two sources of power dissipation in parallel transmission: switching power of each wire and the coupling power between the adjacent wires. Nowadays, with technology scaling, coupling power dominates total power of interconnects. Solutions like introduction of low-K dielectrics between global interconnects layers can reduce coupling capacitance to some degree but are not sufficient. Further, introduction of low-K dielectrics aggravates the problem of wire self-heating [1].

Bus encoding was introduced to reduce toggle count on transmitted data, and hence reduce power dissipation of interconnect. Most existing low-power bus encoding schemes [2][3] typically save energy by fully or partially inverting bus data in selected cycles to reduce bus switching activity.

There are various low-power coding methods for data buses: BI (Bus Invert) code [4] for uncorrelated data patterns, and probability-based mapping [4] for patterns with non-uniform probability densities. BI encoding is one of the simplest coding schemes

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 90–97, 2008.

introduced so far. In [5] a new bus coding scheme, called Partial Bus-Invert (PBI) coding was introduced, where the conventional bus-invert (BI) coding technique is used, but it is applied only to a selected subset of bus lines. Both of these methods try to reduce the number of transitions on wires, and therefore, reduce self energy.

In current and future nanometer scale technology, both self and coupling energies are important. Nowadays, coupling effects are becoming increasingly more pronounced, e.g., $\frac{C_{Coupling}}{C_{Self}}$ = 2.082 for the 130 nm technology, 2.344 for 90 nm, and 2.729 for 65 nm based on ITRS technology and wire geometry parameters for global interconnects routed in the topmost metal layer [6]. OEBI [7] attempts to reduce coupling energy by lowering adjacent transitions. Employing just one of these coding schemes (BI or OEBI) may reduce either self or coupling powers and increase the other.

In [8] a partitioned hybrid encoding (PHE) technique is proposed to optimally partition a bus and apply the most energy-efficient encoding scheme independently to each partition, based on traffic value characteristics, to minimize total bus dynamic energy. This technique considers BI and OEBI techniques. It shows that while BI and OEBI provide average energy reductions of only 5.27%/1.58% and 0.98%/1.97% for data/instruction buses, respectively, for SPEC CPU2k benchmarks, their hybrid encoding technique yields in average energy savings of 22.07%/27.40% for the same traffic and buses.

Control bits in different coding schemes consume additional power which is compensated by power saving in transmitted data. On the other hand, although bus partitioning leads to more power saving compared to conventional coding, it results in more control bits and hence, more power consumption due to considerable number of transitions on these extra bits. We can conclude that although bus partitioning may lead to efficient coding for non-random data, it increases power overhead according to extra control bits. In this paper, we propose a coding scheme based on bus partitioning which attempts to minimize the number of transitions on control bits. This method tries to group together partitions with similar wire transitions behavior, considering self and coupling transitions. To each group of partitions, an extra control bit is added. Although we have increased the number of control bits, their total transitions are reduced. Hence, total power consumption decreases compared to conventional PBI.

In the proposed coding scheme, using proper optimization methods, we search for the efficient number of partitions and their boundaries such that the total power consumption is minimized. Both the number of partitions and their boundaries is determined by PSO [9] optimization algorithm.

PSO consists of a swarm of particles moving in an n dimensional search space of possible problem solutions. Each particle is associated with a velocity. Particles fly through the search space with velocities which are dynamically adjusted according to their historical behaviors. Therefore, the particles track optimum points to search the space and find the solution of the optimization problem [9].

Our proposed coding scheme attempts to find partitions with similar transition behaviors and control them with an extra control bit. Therefore, PSO optimization algorithm tries to control number of partitions jointly to reach maximum power saving. According to insertion of extra control bits in the proposed method and flexibility of

its partitioning algorithm, this method can reduce power consumption of control bits compared to [8]. Therefore, total power saving is improved with respect to [8].

Traditionally, control bits corresponding to each partition are assumed to be located between adjacent partitions. This way, neighboring partitions are decoupled from each other (i.e. there is no coupling transition between them). Since coupling energy constitutes considerable portion of the total energy, it seems beneficial to reduce it through locating control bits efficiently. Our proposed method uses PSO for optimally locating control bits in each partition for obtaining least coupling energy.

To demonstrate effectiveness of the proposed coding scheme, we applied it to a 32-bit instruction bus and a 32-bit data bus in Jpeg encoder and decoder which are synthesized with ODYSSEY [10] synthesis tool. Experimental results are also obtained for a randomly generated data pattern.

The remaining sections are organized as follows; proposed coding scheme is presented in Section 2. Section 3 presents the experimental results. Finally, Section 4 concludes the paper.

2 Proposed Coding Scheme

As mentioned before, the aim of the proposed method is to reduce the transition and coupling powers of bus. For this purpose, the proposed method attempts to find the best partitioning and put the partitions with similar behaviors in the same group. For each of these groups, one extra control bit is added. We will show that although the number of control bits is increased, total power consumption is decreased. In addition to grouping similar partitions, this method tries to efficiently locate control bits of each partition, such that coupling power (and hence total power) decreases. Utilizing both mechanisms (grouping similar partitions and locating control bits) our proposed coding method leads to considerable power reduction of parallel buses. In this section, we will expand our proposed coding method in details.

For reducing the coupling and transition power dissipations, the system supports bus invert (BI) and odd even bus invert (OEBI) coding schemes. Therefore, each partition has two control bits. Efficient number of partitions, their boundaries, those with similar behaviors and position of control bits in each partition are specified with tree PSO optimization algorithms. The first PSO tries to find the best number of partitions and their boundaries, the second one specifies partitions with similar behaviors and groups them together, and finally the third one locates control bits of each partition.

With the first PSO algorithm, in each step of the optimization algorithm a new configuration for the parallel bus is achieved. Bus configuration is defined as follows:

Definition 4.1: Each *configuration* is determined by number of partitions and their boundaries.

Each configured partition can be coded either by BI or OEBI coding scheme, and different partitions are coded independently. For each partition, we chose either of BI or OEBI coding scheme with the higher value of power saving. Power consumption includes both self and coupling components. Fig. 1 represents an example of a partitioned bus with four partitions. The position of control bits dedicated to each partition

is determined by the thirds PSO algorithm, while the extra control bits are located as shown in this figure.

With the second PSO algorithm, in each step of the optimization algorithm (for each bus configuration), our coding scheme tries to find the similar partitions to reduce the activity of control bits. In other words, our method tries to control similar partitions using a common extra bit, and therefore, it reduces the number of transitions on control bits dedicated to each partition. It also proposes a coding technique for each partition to the third PSO algorithm.

Our proposed method may aggregate non-continuous partitions to a group. The aggregation method tries to minimize power consumption. If all aggregated partitions in a group are going to be coded by the same coding scheme (such as OEBI or BI), the extra control bit of the group will be high and the control bits of the first partition in the group indicate the selected coding scheme.

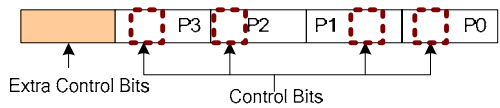


Fig. 1. Position of control bits

As explained before, the final goal of the proposed coding scheme is to reduce the total power consumption. For this purpose, PSO optimization algorithms search for the best bus configuration, which in conjunction with aggregation method and properly locating control bits, leads to minimum power consumption. For each selected configuration, the aggregation method tries to categorize different partitions to some groups in order to save power on control bits.

The proposed algorithm has three phases and each phase implements a PSO algorithm. In the first phase, PSO algorithm tries to find the best bus configuration (including number of partitions and their boundary); in the second phase, another PSO algorithm tries to find the similar partitions and put them in a group (it may categorize partitions to more than one group); and finally in the third phase, with the third PSO algorithm, control bits of each partition are located within that partition such that the total power consumption is minimized. Details of each phase are described as follows.

In the first phase, PSO specifies the partition borders. By changing these values, length of each partition can be adjusted. Therefore, each particle in the first PSO algorithm contains borders of partitions

At first, particles are initialized with some random borders. In fitness calculation step, bus's configuration is specified according to partitions' border, stored in particles. This configuration is sent to the second PSO. After grouping similar partitions together, this configuration is passed to the third PSO. The best particle of the third PSO shows the fitness value of the particle in the first and the second one. After each fitness calculation step, the borders are moved according to velocities of each particle.

In the second phase of the proposed algorithm, another PSO proposes groups of similar partitions. In this phase, each individual of a particle represents a partition. Therefore, the length of particles in this phase is variable and depends on the configuration sent from the first phase. At first, PSO fills the individuals of each particle with

random numbers in the range of 0 to particles' length. In evaluation phase, partitions with the same number are grouped together.

Third phase evaluates partitioning and grouping structure proposed by first and second phases, respectively. It also locates control bits of each partition such that total power consumption is reduced. For this purpose, the third PSO algorithm uses following definition for transition similarity between different partitions:

Definition 4.2: Similar partitions are those partitions which are coded with the same BCS (Best Coding Scheme) in most occurrences.

Definition 4.3: BCS for each partition is the coding scheme which leads to the most power saving compared to other schemes. For example, BCS for a partition can be OEBI with control bits equal to 01.

Evaluation phase of the third PSO is the main part of the proposed coding scheme. It takes bus configuration (from first PSO), grouped partitions (from second PSO), and control bits' positions within each partition (from third PSO) and calculates fitness value. For this purpose, it calculates power consumption for each group of partitions under different *situations* and selects the best situation which leads to the most power reduction compared to the original data (uncoded data). Those partitions that are not included in any group are considered individually.

Possible situations for each group of partitions are listed below:

1. Each partition in the group has its own coding scheme. In other words, same coding technique is not applied to all partitions. Therefore, suitable low power techniques used for each partition are determined by partitions' own control bits.
2. No partition in the group is coded.
3. Odd bits of each partition in the group are inverted.
4. Even bits of each partition in the group are inverted.
5. All bits of each partition in the group are inverted.

Under situation 2, 3, 4, and 5 all partitions are coded similarly. Therefore, they can be controlled by control bits dedicated to one partition (e.g. the first one). In the evaluation phase of third PSO, for each group of partitions power consumption is computed in each of the above situations and the best one is selected. The best situation for each group is the one with the least power consumption.

Fitness value assigned to a bus configuration in the evaluation phase of the second PSO equals to the total power consumption of parallel bus. This value is sum of powers consumed in different groups of the bus (individual partitions are also considered). In addition to power dissipation of the partitions, the proposed method calculates power of control bits for each partition and the extra control bits for each group.

Based on the third PSO's decision, dedicated control bits for each partition may be placed within that partition, or next to MSB or LSB bits of the partition. On the other hand, each partition is placed between two other partitions, except the first one which is only next to one neighboring partition. While calculating the power dissipation of each partition, the proposed method computes the transition and coupling power dissipated in the partition, the power dissipated by coupling transition between neighboring partitions, and transition and coupling power of control bits in the partition.

As explained before, the coding scheme for each partition is selected independently. According to the selected coding scheme, proposed method computes transition and coupling energy of the partition's control bits, but it does not have any knowledge about coding scheme of the neighboring partition. In other words, it cannot compute coupling energy between boundary bit of this partition and that of the neighboring one (note that boundary bit of the neighboring partition may be data or control bit of that partition). To compensate this lack of knowledge, we put an error value instead of coupling power dissipation between neighboring partitions. This value is equal to 1 when the least significant bit (boundary bit) of the partition changes, and is equal to 0.5 when this bit does not change.

3 Experimental Results

We evaluated our proposed coding scheme for JpegDecoder and JpegEncoder testbenches. These benchmarks are written in C++ language and are synthesized into hardware with ODYSSEY synthesis tool. We sampled 1000 data from Data and Address buses which are connected to RAM Manager of the synthesized hardware. For evaluating the proposed method, three different 32*32 pixel images were applied to these benchmarks. These two groups of data (data and address) supply coherent samples.

Fig 2 shows power consumption of the original data (OR) and the coded one resulted from proposed coding scheme (PO). Fig 2(a,b) and Fig 2(c,d) depict power consumption for data and address bus for JpegEncoder, and JpegDecoder, respectively. These figures illustrate average power consumption on three input images. All these figures contain transition, coupling, and total power consumption.

As illustrated in these figures, power savings of the proposed coding scheme on Address bus in both JpegEncoder and JpegDecoder are better than Data bus. The reason is that coherency between addresses is less than that of data. Since consequent data usually refer to adjacent pixels, they are coherent. In contrast, consequent data are not necessarily stored at consequent addresses. Hence, transmitted addresses on address bus are not coherent. As we know, BI and OEBI lead to most power saving in case of coding random data. Consequently power saving for address bus is more than data bus which can be understood from Fig 2.

The coding scheme proposed in [8] enforces dynamic programming by solving recursively for each partition. Each recursion step breaks up the problem and tries to reduce power consumption corresponding to that partition. In [8] each partition in *ith* step is a subset of one of *(i-1)th* partitions. Therefore, employing this recursive method limits possible partitions while proposed partitioning in our coding scheme moves partitions' border without any constraint. Hence, it searches for proper partition in a wider search space compared to [8].

For comparison, we removed mentioned constraint in [8] and employed our partitioning method to it to achieve the modified version of [8]. The modified method leads to better power saving due to its wider search space compared to the original one presented in [8]. Then we compared our proposed method with this modified coding scheme. The main different point in this comparison is that our proposed method adds some extra control bits resulted from aggregated partitions and properly

locates control bits of each partition. These extra control bits do not exist in [8], and it statistically locates control bits of each partition.

Table 1 compares power saving in the proposed method with modified version of [8]. Results are reported in percent with respect to original data. As shown in this table, our proposed method considerably outperforms the modified version of [8]. The proposed coding scheme leads to 46%, 37%, and 40% power saving compared to original data for transition, coupling and total power, respectively. It leads to 7% to 10% more power saving compared to modified version of [8]. Average power savings we gained with respect to the later method are 7% and 10% for coupling and transition power, respectively. This better power saving results from extra control bits added for each group of partitions. As mentioned before, these extra control bits reduce bit transition on conventional control bits of aggregated partitions in each group.

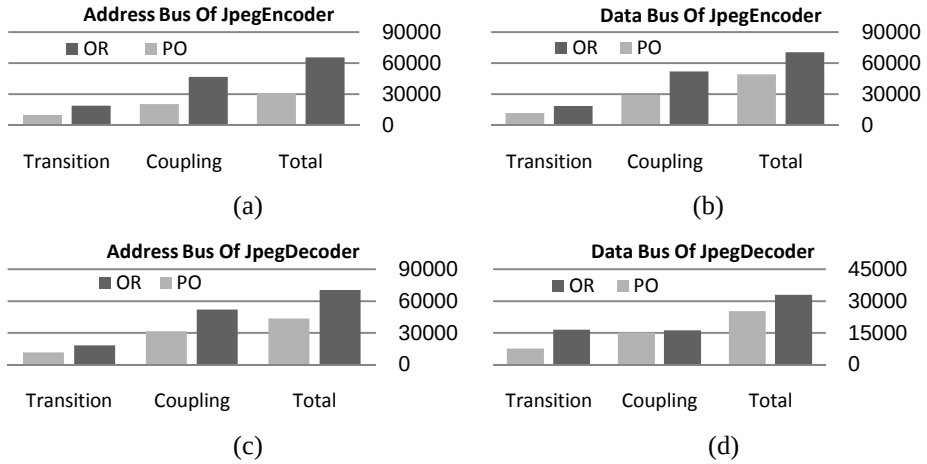


Fig. 2. Power consumption of the original data and the coded one resulted from proposed coding scheme

Table 1. Power saving in the proposed method and modified version of [8]

		Modified [8]			Proposed Method		
		Total	C*	T*	Total	C*	T*
JE	Address	47.77%	49.20%	48.25%	52.76%	56.31%	48.68%
	Data	38.57%	37.67%	37.29%	43.65%	44.09%	44.53%
JD	Address	24.69%	29.58%	12.86%	37.89%	39.31%	36.81%
	Data	19.12%	2.81%	48.54%	23.12%	7.10%	53.73%

* T=Transition, C=Coupling.

4 Conclusions

There are many techniques for reducing the transition and coupling powers in parallel bus interconnection. These methods put extra control bits to manage the behavior of data transitions on parallel bus. These control bits add extra power dissipation which

is compensated by power reduction in coded data. In this paper, we proposed a new coding scheme which tries to reduce power dissipation of control bits. For this purpose, the proposed method employs partitioned Bus Invert and Odd Even Bus Invert coding techniques. This method benefits from Particle Swarm Optimization (PSO) algorithm to efficiently partition the bus. To reduce transition and coupling power of control bits, it finds partitions with similar transition behavior and groups them together. It adds one extra control bit for each group of partitions. By properly managing transitions on control bits of each partition and each group, it reduces total power consumption including coupling power. The proposed coding scheme also tries to properly locate control bits of each partition to reduce coupling power (and hence total power) consumption. We evaluated our method on benchmarks which are synthesized with an object oriented synthesis tool. Address bus and data bus of a JpegEncoder and JpegDecoder are coded for three different images. Experimental results show 40% power saving in coded data compared to original data. We also showed the prominence of the proposed coding scheme compared to other techniques.

References

1. Banerjee, K.: Trends for ULSI Interconnections and Their Implications for Thermal, Reliability and Performance Issues. In: Proceedings of the Seventh International Dielectrics and Conductors for ULSI Multilevel Interconnection Conference, pp. 38–50 (2001)
2. Ghoneima, M., Ismail, Y.: Low power coupling-based encoding for on-chip buses. In: Proceedings of International Symposium on Circuits and Systems, pp. 325–333 (2004)
3. Kim, K.W., Back, K.H., Shanbhag, K.H., Liu, C.L., Kang, S.M.: Coupling-driven signal encoding scheme for low-power interface design. In: Proceedings of the International Conference on Computer-Aided Design, pp. 318–321 (2000)
4. Ramprasad, S., Shanbhag, N.R., Hajj, I.N.: A coding framework for low-power address and data busses. IEEE Transaction on VLSI System 7, 21–221 (1999)
5. Shin, Y., Chae, S.I., Choi, K.: Partial Bus-Invert Coding for Power Optimization of Application-Specific Systems. IEEE Transaction on Very Large Scale Integration Systems 9, 377–383 (2001)
6. Sundaresan, K., Mahapatra, N.R.: Accurate Energy Dissipation and Thermal Modeling for Nanometer-Scale Buses. In: Proceedings of the Eleventh International Symposium on High-Performance Computer Architecture (HPCA-11), San Francisco, pp. 51–60 (2005)
7. Zhang, Y., Yang, J., Gupta, R.: Frequent value locality an value-centric data cache design. In: Proceedings of Architectural support for programming languages and operating systems, pp. 150–159 (2000)
8. Jayaprakash, S., Mahapatra, N.R.: Partitioned Hybrid Encoding to Minimize On-Chip Energy Dissipation of Wide Microprocessor Buses. In: Proceeding of the VLSI Design (VLSID 2007), pp. 127–134 (2007)
9. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings IEEE Intl. Conf. Neural Networks, pp. 1942–1948 (1995)
10. Goudarzi, M., Hessabi, S.: The ODYSSEY Tool-Set for System-Level Synthesis of Object-Oriented Models. In: Hämäläinen, T.D., Pimentel, A.D., Takala, J., Vassiliadis, S. (eds.) SAMOS 2005. LNCS, vol. 3553, pp. 394–403. Springer, Heidelberg (2005)

Efficient Parallel Buffer Structure and Its Management Scheme for a Robust Network-on-Chip (NoC) Architecture

Jun Ho Bahn and Nader Bagherzadeh

Department of Electrical Engineering and Computer Science,
University of California Irvine - Irvine, CA 92697–2625
{jbahn,nader}@uci.edu

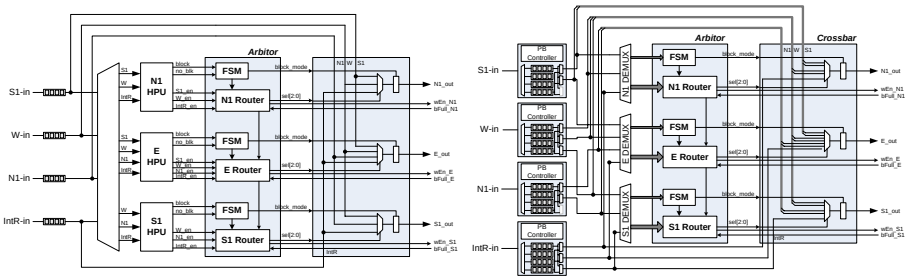
Abstract. In this paper, we present an enhanced Network-on-Chip (NoC) architecture with efficient parallel buffer structure and its management scheme. In order to enhance the performance of the baseline router to achieve maximum throughput, a new parallel buffer architecture and its management scheme are introduced. By adopting an adjustable architecture that integrates a parallel buffer with each incoming port, the design complexity and its utilization can be optimized. By utilizing simulation-based performance evaluation and comparison with previous NoC architectures, its efficiency and superiority are proven. Major contributions of this paper are the design of the enhanced structure of a parallel buffer which is independent of routing algorithms, and its efficient management scheme for the Network-on-Chip (NoC) architecture adopting a minimal adaptive routing algorithm. As a result, the total amount of required buffers can be reduced for obtaining the maximum performance. Additionally a simple and efficient architecture of overall NoC implementation is provided by balancing the workload between parallel buffers and router logics.

Keywords: Network-on-Chip, On-chip network, virtual channel, parallel buffer, router.

1 Introduction

In designing Network-on-Chip (NoC) systems, there are several issues to be considered, such as topology, routing algorithm, performance, latency, and complexity. All these factors are taken into account when the design of an NoC architecture is considered. Regarding routing algorithms, many researchers have developed better performance routing algorithm using oblivious/deterministic or adaptive routing algorithms [1,2,3,4,5,6]. In addition, the adoption of virtual channel (abbreviated to VC) has been prevailing because of its versatility. By adding virtual channels and proper utilization of their channels, deadlock-freedom can be easily accomplished. Network throughput can be increased by dividing the buffer storage associated with each network channel into several virtual channels [4]. By proper control of virtual channels, network flow control

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 98–105, 2008.



(a) base router (b) PB enhanced router

Fig. 1. Micro-architecture of base and PB enhanced router

can be easily implemented [7]. Also to increase the fault tolerance in a network, the concept of virtual channel has been utilized [8,9]. However, in order to maximize its utilization, allocation of virtual channels is a critical issue in designing routing algorithms [10,11].

We proposed a base NoC architecture adopting a minimal adaptive routing algortihm with near-optimal performance and feasible design complexity [6]. Based on this NoC architecture, a new routing-independent parallel buffer structure and its management scheme are proposed instead of VC. As a result, the channel utilization and maximum throughput in performance are improved.

The orginazation of this paper is as following. In the next section, a brief introduction of base NoC architecture adopting a minimal adaptive routing algorithm will be provided. While explaining the proposed parallel buffer (PB) structure and its management scheme, the enhanced NoC architecture including these parallel buffers will be introduced. In order to prove its benefit, several simulation-based evaluation results and comparison with the base NoC architecture will be provided. Finally some conclusions will be drawn.

2 Base Network-on-Chip (NoC) Architecture

We proposed an adaptive routing algorithm and the baseline architecture for a flexible on-chip interconnection [6]. It adopts a wormhole switching technique and its routing algorithm is livelock-/deadloc- free in 2-D mesh. Specifically to eleminate the deadlock situation and simplify the routing algorithm, two disjoint vertical channels are provided instead of using virtual channels. The use of a vertical channel is limited by the direction of delivered data. That is, each vertical channel is denoted by *N1/S1* for east-bounded and *N2/S2* for west-bounded packets, respectively. Also, the data from the internal processing element (PE) or execution unit (EU) connected with router uses separate injection ports, *IntL-in* and *IntR-in*, depending on its direction of target node. As a result, available routing ports are grouped as {*W-in, N1, E-out, S1, IntR-in*} and {*E-in, N2, W-out, S2, IntL-in*} where *N1/N2* or *N2/S2* represent incoming/outgoing ports

Algorithm 1. Pseudo routing algorithm for base router

```
1: local variable: MappediPortFlag[ ] contains whether iPort[ ] is routed or not
2: initialize MappediPortFlag[ ] to 0
3: for oIndex ← N1-out to Int-out do
4:   if oPort[oIndex].Status is active then
5:     traverse data from iPort[oPort[oIndex].RoutediPort] to oPort[oIndex]
6:     decrease oPort[oIndex].RestFlits by 1
7:     MappediPortFlag[oPort[oIndex].RoutediPort] set to 1
8:     if oPort[oIndex].RestFlits is zero then
9:       oPort[oIndex].Status set to inactive
10:    end if
11:  else
12:    for iIndex ← value from left to right at the row of the corresponding outgoing port in
       Tab. 1 do
13:      if MappediPortFlag[iIndex] is 0 then
14:        if iPort[iIndex].HeaderReady is true then
15:          traverse data from iPort[iIndex] to oPort[oIndex]
16:          extract the length of flits and set oPort[oIndex].RestFlits
17:          oPort[oIndex].RoutediPort ← iIndex
18:          MappediPortFlag[iIndex] ← 1
19:        end if
20:      end if
21:    end for
22:  end if
23: end for
```

simultaneously(*-in* an incoming port, and *-out* an outgoing port for the given channel, respectively).

For each set of ports, the routing decision is independently performed. For instance, in the set of east-bounded ports, i.e. {*W-in, N1, E-out, S1, IntR-in*}, incoming ports are routed to each output port depending on port priority shown in Tab. 1. There are two different levels of priority on incoming ports and outgoing ports, respectively. The priority on outgoing ports is given in the order of *N1-out*, *E-out*, *S1-out*, and *Int-out*. Thus, it is organized by starting from north in clockwise direction to the port to EU, *Int-out*, having the lowest priority. The other priority on incoming ports is differently assigned depending on the given outgoing port. For the given outgoing port, the priority on incoming ports is given in the order of clockwise direction starting from the incoming port next to the given outgoing port if the incoming port has a deliverable flit to the given outgoing port. If any incoming port is routed to the outgoing port with higher priority, that port is not considered in routing decision for the outgoing ports with lower priority. Algorithm 1 summarizes a detailed procedure of the routing decision. Based on these operations, the micro-architecture of either *Right* or *Left* router is designed as Fig. 1(a).

3 Enhanced Network-on-Chip (NoC) Architecture with Parallel Buffers

In order to enhance the performance of base NoC architecture, an approach similar to parallel buffer technique of virtual channels is selected as shown in Fig. 1(b). Instead of using dedicated buffers for each port, parallel buffers with

Table 1. Priority assignment on incoming/outgoing ports

outgoing ports	incoming ports
N1-out	*S1-in, W-in, IntR-in*
E-out	*S1-in, W-in, N1-in, IntR-in*
S1-out	*W-in, N1-in, IntR-in*
N2-out	*E-in, S2-in, IntL-in*
S2-out	*N2-in, E-in, IntL-in*
W-out	*N2-in, E-in, S2-in, IntL-in*
Int-out	*N1-in, N2-in, E-in, S1-in, S2-in, W-in*

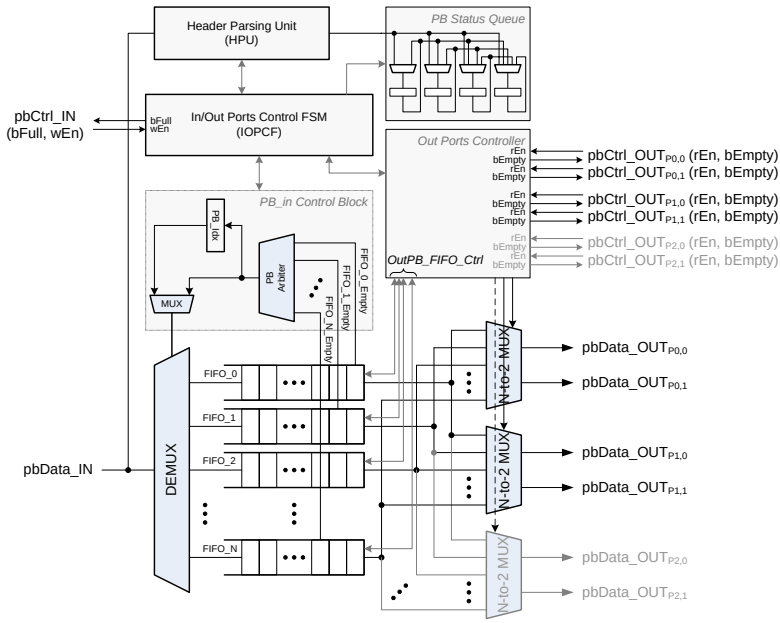


Fig. 2. Proposed parallel buffer structure

small depth queue or FIFO are added in front of each incoming port. The difference from previous approaches with virtual channels is a routing-independent parallel buffer structure, and its efficient management scheme which will be described in detail.

Figure 2 shows a detailed parallel buffer structure applied in the enhanced NoC architecture. To maximize the utilization of channels, multiple outputs from a parallel buffer for each forwarded direction are provided to the routers. By virtue of 2-D mesh topology, the maximum number of forwarded directions is 3. For each forwarded direction, maximum 2 outputs from a parallel buffer are provided. The following example explains how to extract the maximum number of outputs from parallel buffer for each output port.

Let's assume that a parallel buffer with 8 FIFOs at each incoming port is allocated and all FIFOs only in the parallel buffer at the incoming port *W-in*

Algorithm 2. Pseudo routing algorithm for enhanced PB router

```
1: local variable: MappediPortFlag[ ][ ] contains whether iPort[ ][ ] is routed or not
2: initialize MappediPortFlag[ ] to 0
3: for oIndex ← N1-out to Int-out do
4:   if oPort[oIndex].Status is active then
5:     traverse data from iPort[oPort[oIndex].RoutediPort][oPort[oIndex].RoutediPortPB] to
       oPort[oIndex]
6:     decrease oPort[oIndex].RestFlits by 1
7:     MappediPortFlag[oPort[oIndex].RoutediPort][oPort[oIndex].RoutediPortPB] set to 1
8:     if oPort[oIndex].RestFlits is zero then
9:       oPort[oIndex].Status set to inactive
10:    end if
11:  else
12:    for iIndex ← value from left to right at the row of the corresponding outgoing port in
       Tab. 1 do
13:      for iPBIndex ← value from the order of PB queue do
14:        if MappediPortFlag[iIndex][iPBIndex] is 0 then
15:          if iPort[iIndex][iPBIndex].HeaderReady is true then
16:            traverse data from iPort[iIndex][iPBIndex] to oPort[oIndex]
17:            extract the length of flits and set oPort[oIndex].RestFlits
18:            oPort[oIndex].RoutediPort ← iIndex
19:            oPort[oIndex].RoutediPortPB ← iPBIndex
20:            MappediPortFlag[iIndex][iPBIndex] ← 1
21:          end if
22:        end if
23:      end for
24:    end for
25:  end if
26: end for
```

contain packets. Also the packets occupying FIFOs in the parallel buffer at *W-in* port, arrived at different times. The destination of each packets occupying each FIFO in the parallel buffer is *E*, *NE*, *N*, *SE*, *SE*, *S*, *NE*, and *S* in the order of $\{PB_0, PB_1, PB_2, PB_3, PB_4, PB_5, PB_6, PB_7\}$ where PB_i represents the i-th FIFO in the given parallel buffer. Also the order of arrival time for each packet is $\{PB_1, PB_3, PB_0, PB_2, PB_4, PB_6, PB_5, PB_7\}$. If each FIFO in the parallel buffer is grouped in the order of arrival time and its available routing direction, the resultant groups of FIFOs are $\{PB_1, PB_2, PB_6\}$ for *N-out*, $\{PB_1, PB_3, PB_0, PB_4, PB_6\}$ for *E-out*, and $\{PB_3, PB_4, PB_5, PB_7\}$ for *S-out*. Because no other incoming ports than *W-in* have deliverable data, the routing decision is performed only on the port *W-in*, According to the described priority in Tab. 1, the outgoing port *N-out* is the first one to be considered. For *N-out* outgoing port, the packet stored in PB_1 will be selected. For *E-out* outgoing port, the packet stored in PB_3 will be chosen because PB_1 is already occupied by the outgoing port *N-out* with higher priority. Finally to *S-out* outgoing port, the packet stored in PB_4 will be forwarded because the earlier packet in PB_3 is already served for *E-out*. Therefore, instead of searching all the entries in each group, the first 2 entries are sufficient for checking the available incoming packet for the routing decision. Algorithm 2 summarizes a detail routing procedure for the enhanced PB router.

Different from the conventional VC approaches, the operation of the proposed parallel buffer is no longer dependent on neighboring routers. By autonomous management of a parallel buffer depending on outgoing port read requests, the parallel buffer can be simply assumed as single FIFO. That is, from the

previous neighboring router, the parallel buffer is recognized as a single FIFO with ordinary interfaces such as `bFull` (buffer fullness) or `wEn` (write enable). Therefore, the only task of this parallel buffer is to store the incoming packets and manage their occupancy with respect to their destination and read operations depending on routing decision. In order to manage the empty FIFOs in the given parallel buffer, as illustrated in Fig. 2, simple logic circuits are added in *PB_in Control Block*. With given empty signals from all input FIFOs, one of empty FIFO indices is chosen which controls the path of storing incoming *flit* into the corresponding FIFO. Simultaneously to control the incoming packet in *flit*, the header parsing unit (HPU) and associated control unit (IOPCF) are needed.

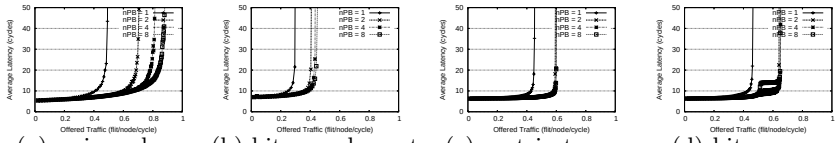
In the proposed parallel buffer structure, every two outputs among the allocated FIFOs in the parallel buffer are chosen and forwarded to the inputs of routing decision logic for the corresponding outgoing port. The parallel buffer controller manages the history of arrival packets and their residence in FIFOs at the parallel buffer, and groups of in-use FIFOs based on its outgoing direction. For this purpose, in the parallel buffer controller, header parsing unit (HPU) for incoming packets is required. By moving the location of header parsing unit which is originally placed in the router logic as Fig. 1(a), the critical path in the enhanced router can be reduced. Because the performance of FIFOs is much faster than the one for the router logic with HPU [6], it results in balancing the workload of each blocks with respect to the timing. Therefore, for the enhanced NoC architecture, the better timing performance in real implementation can be expected.

4 Evaluation of the Performance in the Enhanced NoC Architecture

4.1 Evaluation Environment

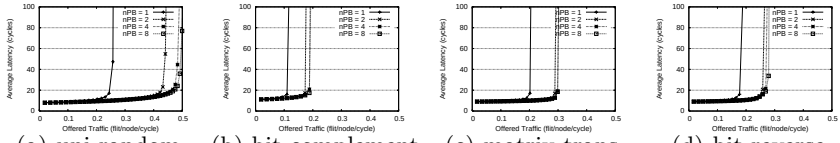
In order to evaluate the performance of the base NoC architecture, a time-accurate simulation model was implemented in SystemC. By comparing with different routing algorithms, its competitive performance has been evaluated. In this paper, by adding parallel buffers with efficient management scheme, overall performance increment is expected. Therefore, the parallel buffer with proposed management scheme is modeled similarly in SystemC. And the previous FIFO module is swapped with this parallel buffer module.

All the network simulations were performend using 100,000 cycles with 4 commonly used traffic patterns such as uniform random, bit-complement, matrix-transpose traffic, and bit-reverse traffic. Two different sizes of 2-D mesh topologies based on 4×4 and 8×8 were studied. Also the number of FIFOs in the parallel buffer per incoming port is varied. However, the depth of FIFO in the parallel buffer is fixed as 4 and 4-flit long packets are used. For the measurement of throughput and adjusting incoming traffic, the standard interconnection network measurement technique [1] was adopted.



(a) uni-random (b) bit-complement (c) matrix-trans (d) bit-reverse

Fig. 3. Performance in 4×4 mesh



(a) uni-random (b) bit-complement (c) matrix-trans (d) bit-reverse

Fig. 4. Performance in 8×8 mesh

4.2 Simulation Results and Their Analysis

Throughout the SystemC simulations with various traffic patterns and two different network topologies, many experimental results of PB enhanced NoC architecture are well collected. With these collected data, the plots of average latency vs. offered traffic are drawn in Fig. 3 and Fig. 4 for 4×4 and 8×8 mesh, respectively. In 4×4 mesh network, both uniform random and bit-reverse traffic patterns show the notable increase of maximum throughput, approximately 25% and 19%, respectively. In 8×8 mesh network, uniform random, matrix-transpose, and bit-reverse traffic patterns show the noticeable improvement, around 28%, 28%, and 18%, respectively. However, for bit-complement traffic pattern in 4×4 and 8×8, the improvement of performance seems to be minor. The reason of minor improvement for bit-complement traffic pattern is because it has relatively lower flexibility in choosing routing paths from source to destination and most of the traffic patterns concentrate on the central region for a given mesh, resulting in severe routing contention and blocking similar with the analysis in [11]. Also as the size of 2-D mesh topology increases, the effect of parallel buffer in improving the performance is growing because the increased size provides much higher degree of flexibility in routing paths.

As shown in [6], the previous base NoC architecture reaches up to 0.4 offered traffic at uniform random traffic pattern in 8x8 mesh network even when infinite buffers are allocated between links. However, in the new parallel buffer adopted NoC architecture, the performance already outperforms even when two-FIFO parallel buffer per incoming port are applied as shown in Fig. 4(a). Furthermore, by applying four-FIFO parallel buffer per incoming port, the maximum throughput in 8x8 mesh reaches up to 0.45 (about 13% improvement). With comparison to the base NoC architecture, four-FIFO parallel buffer per incoming port achieves an optimal performance benefit. Also comparing with general virtual channel application [3] where at least 8 virtual channels per physical channel are required to get the nominal performance and resolve deadlock problem, the proposed NoC architecture with parallel buffers has its own benefit.

5 Conclusion

We proposed a new parallel buffer structure and its management scheme, as well as its optimal micro-architecture. By applying this proposed parallel buffer to the previous base NoC architecture, noticeable performance improvement was observed using simulation of various traffic patterns. Even though the deadlock-freedom is realized by providing disjoint vertical channels instead of using virtual channels which is a general approach for this purpose, notable performance benefit can be extracted by adding parallel buffers with smaller number of FIFOs. Also by moving the header parseing unit into the parallel buffer controller, the timing balance between parallel buffer and router logic can be obtained at micro-architecture level.

References

1. Dally, W.J., Towles, B.: Principles and Practices of Interconnection Networks. Morgan Kaufmann, San Francisco (2004)
2. Sullivan, H., Bashkow, T.R., Klappholz, D.: A Large Scale, Homogeneous, Fully Distributed Parallel Machine. In: ISCA 1977, pp. 105–117. ACM Press, New York (1977)
3. Seo, D., Ali, A., Lim, W., Rafique, N., Thottethodi, M.: Near-Optimal Worst-Case Throughput Routing for Two-Dimensional Mesh Networks. In: ISCA 2005, pp. 432–443. ACM Press, New York (2005)
4. Dally, W.J., Seitz, C.L.: Deadlock-Free Message Routing in Multiprocessor Interconnection Networks. IEEE Trans. Computer C-36(5), 547–553 (1987)
5. Glass, C.J., Ni, L.M.: The Turn Model for Adaptive Routing. J. ACM 31(5), 874–902 (1994)
6. Bahn, J.H., Lee, S.E., Bagherzadeh, N.: On Design and Analysis of a Feasible Network-on-Chip (NoC) Architecture. In: ITNG 2007, pp. 1033–1038. IEEE Computer Society, Washington (2007)
7. Dally, W.J.: Virtual-Channel Flow Control. IEEE Trans. Parallel and Distributed Systems 3(2), 194–205 (1992)
8. Boppana, R.V., Chalasani, S.: Fault-Tolerant Wormhole Routing Algorithms for Mesh Networks. IEEE Trans. Computers 44(7), 846–864 (1995)
9. Zhou, J., Lau, F.C.M.: Adaptive Fault-Tolerant Wormhole Routing with Two Virtual Channels in 2D Meshes. In: ISPAN 2004, pp. 142–148. IEEE Computer Society, Los Alamitos (2004)
10. Vaidya, A.S., Sivasubramaniam, A., Das, C.R.: Impact of Virtual Channels and Adaptive Routing on Application Performance. IEEE Trans. Parallel Distributed Systems 12(2), 223–237 (2001)
11. Rezazad, M., Sarbazi-azad, H.: The Effect of Virtual Channel Organization on the Performance of Interconnection Networks. In: IPDPS 2005, p. 264.1. IEEE Computer Society, Washington (2005)

Integration of System-Level IP Cores in Object-Oriented Design Methodologies

Shoaleh Hashemi Namin and Shaahin Hessabi

Department of Computer Engineering
Sharif University of Technology Tehran, Iran
namin@ce.sharif.edu, hessabi@sharif.edu

Abstract. IP core reuse is popular for designing and implementing complex systems, because reuse of already provided blocks decreases design time and so diminishes productivity gap. Moreover, as system-level design methodologies and tools emerge for embedded system design, it is useful to have a shift from Register Transfer Level to system-level models for IP cores employed for implementation of hardware parts of the system. In this paper, we propose a C++ model for hardware IP cores that can be adopted as a standard for delivering IPs at a high level of abstraction, suitable for object-oriented system-level design methodologies. Next, we extend our system-level synthesizer in order to integrate IP cores automatically in a system architecture model generated by the synthesizer. Finally, we validate the extended synthesizer by designing and implementing systems with proposed C++ IP cores in our extended system-level design environment.

Keywords: C++ IP Cores, System-Level Synthesizer, Embedded System.

1 Introduction

In order to solve the productivity gap problem, three approaches have been proposed: (1) Platforms, (2) Reuse and (3) Synthesis [1]. System-level design methodologies are in synthesis approach category, because in these methodologies the SoC chip will be synthesized from a high-level functional description [1].

In system-level design, the key to coping with the complexities involved with SoC design is the reuse of Intellectual Property (IP) cores [2]. In this paper, our main goal is to propose a system level model in C++ for existing IP Cores modeled in Register Transfer Level (RTL) in HDL languages. In order to validate these models, we design and implement a system in our previously extended ODYSSEY environment [3], where we had defined a methodology to add IP core usage capability to ODYSSEY methodology which proposes an object-oriented and system-level design method for embedded systems [4]. We facilitate IP integration by automating wrapper generation which leads to an automated integration process. We will show the process of integrating this model in our specific high level design methodology. However, depending on the specific requirements of each different design methodology, the details of integration process may differ.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 106–114, 2008.

After adopting system-level design methodologies, tools, and environments for SoCs and embedded system design, the lack of high-level models for IP cores posed itself as a problem. Consequently, IP core developers made their efforts to deliver RTL IP cores with their equivalent models at high-level of abstraction to be used in system-level design and verification processes.

There are few system-level design environments with IP reusability feature. In most of them, a system is implemented with IP cores as a case study, without presenting an integration methodology.

In [5], an industrial test case is designed and implemented. A methodology is defined for the integration of CSELT VIP Library, which is a library of customizable soft IP cores, written in RTL VHDL. IP cores are presented only in RTL, and IP integration is done manually for a specific application. We will present our methodology, where IP cores are presented in all abstraction levels for system design and verification, and IP integration is accomplished automatically.

A compiler that takes algorithms described in MATLAB and generates RTL VHDL is presented in [6]. IP integration capability has been added to the compiler, and IP cores have been integrated automatically into the synthesized designs. The difference between this work and our proposed method is that our methodology handles embedded systems composed of both software and hardware parts, but in [6] the compiler has been defined as a hardware generation tool. Moreover, our specification language is C++, while theirs is MATLAB.

BALBOA [7] is a component composition framework. In this framework, an embedded system is modeled by assembling IPs described in SystemC. This framework uses bottom-up design approach, but our design environment and tool benefit from a top-down design approach.

The remainder of this paper is organized as follows. In Section 2, we introduce C++ classes that are used to model hardware IP components in system-level. In Section 3, we review the system-level synthesizer tool. We describe our modified version of synthesizer tool for automating core integration in synthesized embedded system in Section 4. Validating procedure for the modified tool and system synthesis experiments are given in Section 5, and finally, Section 6 concludes the paper.

2 Core Classes

For designing and implementing embedded systems in an object-oriented environment, every function can be implemented as a method of a class (a method of an object). Suppose that we want to implement a method as the hardware part of an embedded system and intend to implement it with reusable IP cores. In this case, it is useful to have a C++ model for hardware IP cores in a system-level design flow, because with these models that correspond to gate-level or RTL hardware IP cores, we can simulate the whole system at a higher level of abstraction using our design flow. Therefore, we propose to model each hardware IP core by a C++ class. The core functionality is modeled as a method of a class, and the inputs and outputs of the core are modeled as the class attributes. In Fig. 1, a core and its C++ model are presented. We call these classes as core classes. As seen in Fig. 1, run() method performs IP core functionality and Data-in and Data-out are the inputs and outputs of IP core, respectively. On the other hand, one essential characteristic of IP cores is source code

security, so we should find a way to code these classes such that the code cannot be seen or changed. Furthermore, we want to have a library of IP cores that can be used in both C++ (our system-level description language) and SystemC (description language of co-simulation model generated by our system-level synthesizer, which is described later) environments. For these reasons, we export core classes from a Dynamic Link Library (dll). This way, run() method that performs core functionality is in dll, and thus secure. The question may arise on how to synthesize these classes, which are exported from dll, from C++ to RTL model. The answer is that we suppose there is an RTL model for these core classes delivered by vendors, so we do not have to synthesize them from C++ to sysemC RTL model.

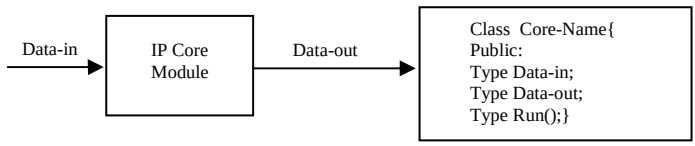


Fig. 1. C++ model of IP cores

Because of using dll, core classes are already compiled with a certain compiler for a certain range of processors. As a result, we can categorize them as hardcore classes or hard system-level IP cores.

3 System-Level Synthesizer

Our synthesizer is based on a system-level design methodology, named ODYSSEY (Object-oriented Design and sYntheSiS of Embedded sYstems) [8], so in this section we first introduce this methodology, and then discuss the single processor synthesis flow.

ODYSSEY synthesis methodology starts from an object-oriented model, and provides algorithms to synthesize the model into an ASIP and the software running on it. The synthesized ASIP corresponds to the class library used in the object-oriented model, and hence, can serve other (and future) applications that use the same class library. This is an important advantage over other ASIP-based synthesis approaches since they merely consider a set of given applications and do not directly involve themselves with future ones.

One key point in ODYSSEY ASIP is the choice of the instruction-set: methods of the class library that is used in the embedded application constitute the ASIP instruction-set. The other key point is that each instruction can be dispatched either to a hardware unit (as any traditional processor) or to a software routine.

An OO application consists of a class library, which defines the types of objects and the operations provided by them, along with some object instantiations and the sequence(s) of method calls among them. We implement methods of that class library as the ASIP instructions, and realize the object instantiations and the sequence of method calls as the software running on the ASIP. Therefore, our core class definition is consistent with OO modeling approach in ODYSSEY methodology.

Fig. 2 shows the synthesis flow for a single-processor target. The entire process is divided into two layers. We consider the upper layer as system-level synthesis; this layer takes the system model and produces the software, along with the hardware architecture in a mixture of structural and behavioral modeling styles. The lower layer is considered as downstream synthesis; it takes the above-mentioned hardware and software partitions, and produces the gate-level hardware and object-code software.

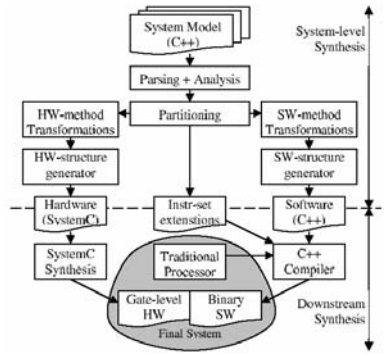


Fig. 2. The ODYSSEY single-processor synthesis flow [8]

4 Core Classes Integration Methodology

According to single processor synthesis flow, hardware methods in downstream synthesis layer are synthesized to gate-level hardware. We propose reusing gate-level IP cores instead of behavioral synthesis to implement certain hardware methods. This way, it is not important to have a synthesizable SystemC model of the corresponding hardware methods.

To reduce design time as mentioned in Section 1, it is useful to have capability of core reusability at higher levels of abstraction during system design and simulation. In his paper, we focus on system-level synthesis and modify system-level synthesis tool to provide the capability of using core classes in the design flow.

In system-level synthesis (upper layer in Fig. 2), input program and class library are parsed and analyzed to extract class-inheritance hierarchy, symbol table, and definitions of methods. At the next stage, these methods are assigned to either hardware or software partitions (which is done by partitioning unit shown in Fig.2), and then the method definitions are transformed to suit their assigned partitions (Transformations units). It should be mentioned that HW/SW partitioning is currently performed manually in ODYSSEY synthesis flow.

Transformed methods in each partition are then passed to their corresponding structure generator unit that appropriately assembles them together. The resulting hardware (in SystemC) and software (in C++) can be simulated together, as the co-simulation of the post-synthesis system, in any environment that supports the SystemC class library. Fig. 3 illustrates the co-simulation model generated from C++ program by system-level synthesizer. The input program is given as a set of header and program files that define the class library as well as the main() function (the left

hand side of Fig. 3), where the objects are instantiated and the sequence of method calls among them is specified. The output is a synthesizable processor architecture comprised of some hardware units and a traditional processor core, along with a software architecture containing a set of software routines and a single thread_main() function (the right hand side of Fig. 3). The implementations of hardware methods are put in the hardware modules in the middle of the OO-ASIP box in Fig. 3, whereas the implementations of software methods are put in the "traditional processor" module. The input main() function is converted to the thread_main() routine in the processor software. The objects' data are put in an object-identifier-addressable memory unit, referred to as object management unit or OMU [8].

As seen in Fig. 3, each HW method is synthesized to a SystemC module. General structure of SystemC modules generated by synthesizer for HW methods. As illustrated, the SystemC module is composed of the net_reciever() and the main() functions. The net_reciever() function is an interface between this method and other ones (either hardware or software), and the main()performs method's main functionality. This function is generated by synthesizer after processing C++ code of the hard ware method, and C++ statements are replaced with some macros introduced in the synthesizer.

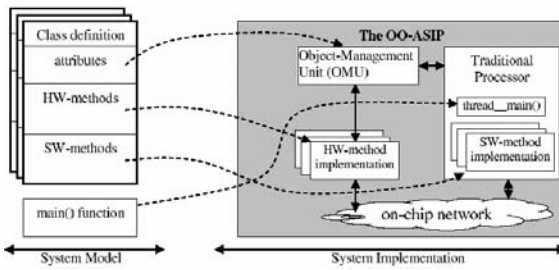


Fig. 3. The big picture of transformations in the system-synthesis process [8]

4.1 Wrapper Classes

Since run() method of a core class used in a program is in dll format and this method does not have a body in the program, the synthesizer cannot detect run() method of the class. This problem is solved by defining a new method, inside which run() method of the core class is called, and therefore, this new method can be detected by the synthesis tool at the system-level. This method does the same work as run() method, while it has a body which does not exist in the original run() method. This definition results in a new class that contains our core class. This class is like wrapper for core class in the system-level. Therefore, we call it wrapper class, which is defined in Fig. 4.

As Fig. 4 illustrates, the wrapper class declaration is similar to the core class declaration, but wrapper-run() method of the former class contains an object of the later one. As seen in Fig. 4, wrapper-run() method passes user data (Wrapper-in) to Data-in of the core object, calls run() method of this object, and finally takes core object Data-out and passes it to Wrapper-out for user.

4.2 SystemC Module of Wrapper Method

Since core and wrapper classes are introduced for implementing hardware methods, they are synthesized as hardware methods by synthesizer. Synthesizer generates a SystemC module for wrapper-run() method of the wrapper class. To generate main() function for wrapper-run() SystemC module, synthesizer processes wrapper-run() C++ code shown in Fig. 4, and replaces all accesses to the object attributes with special macros (described later). But as mentioned before, since our core class is exported from dll, this process leads to an incorrect result and therefore, SystemC co-simulation model generated by synthesizer is not correct when we use core and wrapper classes.

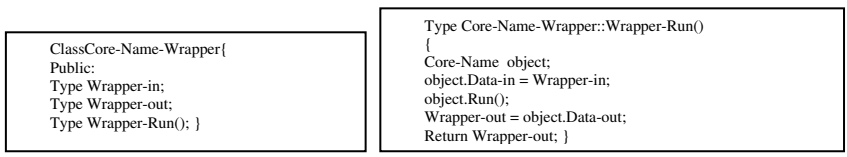


Fig. 4. Definition of (a) wrapper class, (b) wrapper-run() method

In order to solve this problem, we propose using core object in hardware method SystemC module without any change or manipulation; i.e., using dll functions, like run() method of the core class in SystemC co-simulation model. The model proposed for synthesized wrapper-run() method is illustrated in Fig. 5. OBJECT_ATTR_READ() and OBJECT_ATTR_WRITE() macros are defined for reading and writing object attributes, and HW_RETURN_VAL() is defined for returning values from hardware methods.

As Fig. 5 illustrates, to access wrapper object attributes, synthesizer replaces these access statements with their macros, but to access core object attributes, synthesizer should not use the macros.

```
void main() {
   Core-Name object;
   object.Data-in=OBJECT_ATTR_READ (self, Core-Name-Wrapper __ Wrapper-in _index );
   object.run()
   OBJECT_ATTR_WRITE (self, Core-Name-Wrapper __ Wrapper-out _index , object.Data-out);
   HW_RETURN_VAL(OBJECT_ATTR_READ (self, Core-Name-Wrapper __ Wrapper-out _index ), ret_val_len
); }
```

Fig. 5. main() function of wrapper-run() SystemC module

In order to generate proposed model shown in Fig. 5 automatically for wrapper-run() hardware methods, we modified our system-level synthesizer as follows.

4.3 System-Level Synthesizer Modification

Our synthesis tool is an object-oriented tool and consists of three major objects: an *input parser* object which reads the input file and produces a parse tree along with

other necessary data structures for further processing [4]; *per partition synthesizer* object of types FU_synthesizer and sw_synthesizer. FU_synthesizer and sw_synthesizer are the main objects responsible for synthesizing hardware and software methods respectively. They operate on individual method definitions and transform them to suit hardware or software implementation according to their assigned partition; an *output writer* object that assembles all the hardware and software methods synthesized by the synthesizer objects and writes out the co-simulation model of the OO-ASIP [4].

In order to automate core and wrapper classes' integration in SystemC co-simulation model, we modified the synthesis tool. In order to create proposed model for wrapper-run() hardware method, we modified FU_synthesizer object such that it generates our proposed model for the main() function of the synthesized wrapper-run() methods. Moreover, correct operation of co-simulation model which consists of core and wrapper classes as well as user-defined classes, requires a header file containing declaration of the core classes so we changed output writer object to generate this header file automatically.

The big picture of the extended synthesis tool with IP core reusability and automatic IP integration capability is shown in Fig. 6. As seen in this figure, if the designer has a dll which contains core classes' definitions, and these classes are exported from dll, he/she can write wrapper classes for the core, and use them in C++ system model. Then, ODYSSEY synthesis tool only needs two files to generate the co-simulation model automatically: a file containing names of core classes, and a file containing names of wrapper classes.

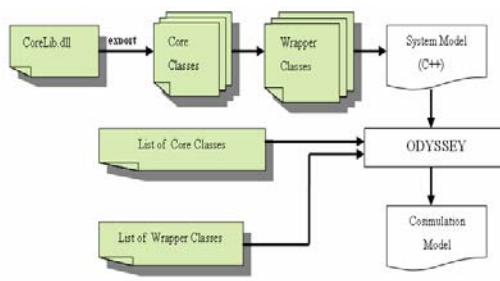


Fig. 6. Extended synthesis tool

5 Experimental Results

Proper operation of the synthesis tool was verified in [8]. The co-simulation capability of generated hardware/software systems ensures us that the system synthesis is correctly performed on each test case, by running the pre- and post-synthesis programs and comparing their outputs.

To test the modified synthesizer, we introduced some core classes and their wrapper classes, and implemented some applications in C++ with these classes. Then, we synthesized C++ programs with the modified synthesizer, ran the pre- and post- synthesis programs and compared their outputs.

To validate our system-level IP integration methodology in ODYSSEY synthesis tool, we used the following case studies:

1. A system which contains one nearly complex core: 2-D Inverse Discrete Cosine Transform containing IDCT core class.
2. A complex system which contains one core, accompanied by other user-defined parts: Jpeg Decoder containing IDCT core class.
3. A system which contains two cores: the Factorial function containing subtracter
4. and multiplier core classes.

Table 1 shows the details of the implemented systems. The case studies were chos en such that different approaches for calling the IP core; i.e., by a software method (SW method) or by a hardware method (HW method), have been examined. Table 3 also presents case studies' complexity in terms of their number of HW and SW methods. In this table, synthesized method refers to user-defined method which is synthesized in an ordinary way (ODYSSEY conventional synthesis approach), while core method refers to wrapper-run() method which should be synthesized by our proposed synthesis approach.

Table 1. The details of the implemented systems

	# SW methods	**# HW methods**		**Core called by**
		Synthesized method	*Core method*	
2-D IDCT	1	-	1	SW method
JPEG Decoder	13	18	1	HW method
Factorial	1	-	2	SW method

Table 2. Simulation results

2-D IDCT Function Implementation	**# Execution Clock Cycles**
IP Core IDCT	**175**
ODYSSEY IDCT	**516**

Table 3. Implementation results

2-D IDCT Function Implementation	**Number of Slices**	**Total gate count**	**Maximum Clock Frequency**
IP Core IDCT	2406 out of 15360 (15%)	59,132	167.497MHz
ODYSSEY IDCT	5176 out of 15360 (33%)	65,757	26.130MHz

As mentioned above, we designed and implemented a JPEG decoder with 19 hardware methods and 13 software ones. We designed and implemented this system in two ways: first we did not use any cores for hardware methods; then, we used a 2-D IDCT core to implement one of the 19 hardware methods. In the first system, we synthesized 2-D IDCT with systemC compiler to generate its RTL model, while in the second system, we used Xilinx RTL model of 2-D IDCT core, directly. Since these systems differ only in 2-D IDCT hardware method, we compare these 2-D IDCT hardware methods in terms of area and speed.

In Tables 2 and 3, IP Core IDCT refers to an IDCT hardware method implemented with Xilinx 2-D IDCT core, while ODYSSEY IDCT refers to the one synthesized with systemC compiler. As shown in these tables, hardware method implemented with IP core is more optimized than the one synthesized from a systemC high level model in terms of speed and area (which is an expected result).

We validated the modified synthesizer tool and the automation of IP integration by modeling the above systems and verifying their pre- and post-synthesis models with certain data and comparing their outputs.

6 Conclusions

Integration of hard core IPs in ODYSSEY methodology has been validated by designing, verifying, and implementing systems with cores in ODYSSEY design environment, but in this paper our focus was on the system-level IP integration issues.

Since IP core vendors do not deliver C++ models for their cores, we created these models ourselves and used them in our ODYSSEY design environment in order to implement the target systems.

In this paper, we introduced a C++ model that can be adopted as a standard model for delivering system-level IP cores. We employed this model in our object-oriented system-level design environment to illustrate validity of the proposed model. Moreover, we facilitated IP integration by automating integration process in SystemC co-simulation model and defining wrapper classes that can be described easily by designer.

References

1. Gajski, D.D., Wu, A.C.H., Chaiyakul, V., Mori, S., Nukiyama, T., Bricaud, P.: Essential Issues for IP Reuse. In: Proc. ASP-DAC 2000, Yokohama, Japan, pp. 37–42 (2000)
2. Hekmatpour, A., Goodnow, K.: Standards-compliant IP Design Advantages, Problems, and Future Directions. In: DesignCon East 2005, Worcester, Massachusetts, USA (2005)
3. Hashemi Namin, S., Hessabi, S.: An IP Integration Metodology in an Object-Oriented design environment for embedded systems. In: 12th Annual International CSI Computer Conference (CSICC 2007), Tehran, Iran (in Persian, 2007)
4. Goudarzi, M.: The ODYSSEY methodology: ASIP-Based Design of Embedded Systems from Object-Oriented System-Level Models. PhD Thesis, Sharif University of Technology (2005)
5. Filippi, E., Lavagno, L., Licciardi, L., Montanaro, A.: Intellectual Property Re-use in Embedded System Co-design: an Industrial Case Study. In: 11th International Symposium on Systems Synthesis, Hsinchu, Taiwan, pp. 37–42 (1998)
6. Haldar, M., Nayak, A., Choudhary, A., Banerjee, P.: A System For Synthesizing Optimized FPGA Hardware From MATLAB. In: ICCAD 2001, San Jose, CA, USA, pp. 314–319 (2001)
7. Shukla, S.K., Doucet, F., Gupta, R.K.: Structured Component Composition Frameworks for Embedded System Design. In: Sahni, S.K., Prasanna, V.K., Shukla, U. (eds.) HiPC 2002. LNCS, vol. 2552, pp. 663–678. Springer, Heidelberg (2002)
8. Goudarzi, M., Hessabi, S.: The ODYSSEY tool-set for system-level synthesis of object-oriented models. In: Hämäläinen, T.D., Pimentel, A.D., Takala, J., Vassiliadis, S. (eds.) SAMOS 2005. LNCS, vol. 3553, pp. 394–403. Springer, Heidelberg (2005)

Polymorphism-Aware Common Bus in an Object-Oriented ASIP

Nima Karimpour Darav and Shaahin Hessabi

Department of Computer Engineering,
Sharif University of Technology, Tehran, Iran
karimpour@alum.sharif.edu, hessabi@sharif.edu

Abstract. Polymorphism and dynamic binding of processing units to methods are key features in object-oriented (OO) processors. Since Network-on-Chip is not available for most platform implementation, common bus can be a good approach to interconnect processing units of an OO processor. In this paper, we present a network protocol with little overhead for common bus that does not violate the polymorphism method calls in an OO processor. Moreover, we show the appropriateness of our solution by presenting a real-world application.

Keywords: Polymorphism, OO processors, Arbiter, Round-robin.

1 Introduction

Object-Oriented (OO) modeling attempts to gain reusability and robustness in the field of software design by means of utilizing abstraction, encapsulation, modularity and hierarchy. These features are what embedded system designers might be motivated to exploit in order to attain shorter time-to-marker. On the other hand, Application Specific Instruction set Processors (ASIPs), present a new trend in the field of embedded system design. They exploit the advantages of traditional processors and Application Specific Integrated Circuits (ASICs) to acquire best features of both.

Blending OO modeling and ASIP-based approach is what ODYSSEY (Object-oriented Design and sYntheSiS of Embedded sYstems) methodology proposes for designing embedded systems [1]. Since the idea of polymorphism is to dynamically bind the related method implementation to the called method at runtime, this feature of OO modeling may enforce extra overhead to implement. Therefore, polymorphism is a crucial feature in Object-Oriented Application Specific Instruction Processors (OO-ASIPs) [2]. In order to avoid the implementation overhead of polymorphism method calls, Reference [3] proposes a solution based on Network-on-Chips. Notwithstanding, the NoC approach is feasible when our platform implementation allows exploiting it; otherwise, a simple bus architecture may be a good candidate. But when we utilize a simple bus instead of an NoC in the structure of OO-ASIPs, the implementation of polymorphism and bus arbitration are the problems we confront. In the case of sequential method calls in an OO-ASIP, there is no more than one request at a time to access the bus, so a simple logic is sufficient for managing the bus accessibility. In the case of concurrent method invocation, the mentioned problems are so crucial that earlier proposed techniques such as [4], [5] and [6] cannot be employed,

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 115–122, 2008.

because a packet transmitter cannot determine whether its packet receiver is idle or busy. Moreover, since methods are called in polymorphic scheme, the packet receiver cannot be determined by the transmitter at runtime. These indeterminisms confuse a regular bus arbiter in deciding whether a transaction should be aborted or resumed.

In this paper, we propose a protocol to cope with the mentioned problems. We illustrate that our solution needs no clock synchronization between a packet transmitter and its packet receiver. This feature is useful to manage power consumption of OO-ASIPs. Moreover, we present a centralized arbiter as well as an arbitrating algorithm with little overhead to prevent bus contentions and the starvation problem.

The next section is a glance at ODYSSEY methodology. We describe our protocol in Section 3 and present experimental results for a simple application in Section 4. Sections 5 and 6 present related work and conclusions.

2 ODYSSEY Methodology

Software accounts for 80% of the development cost in today embedded systems [7] and object-oriented design methodology is a well-established methodology for reuse and complexity management in software design community. In addition, the OO methodology is inherently developed to support incremental evolution of applications by adding new features to (or updating) previous ones. Similarly, embedded systems generally follow an incremental evolution (as opposed to sudden revolution) paradigm, since this is what the customers normally demand. Consequently, we believe that OO methodology is a suitable choice for modeling embedded applications, and hence, this path is followed in ODYSSEY.

ODYSSEY advocates programmable platform or ASIP-based approach, as opposed to full-custom or ASIC-based philosophy of design. Programmability, and hence programmable platforms, is one way to achieve higher volumes by enabling the same chip to be reused for several related applications, or different generations of the same product. Moreover, programming in software is generally a much easier task compared to designing and debugging working hardware, and hence, programmable platforms not only reduce design risk, but also result in shorter time-to-market [8].

ODYSSEY synthesis methodology starts from an OO model, and provides algorithms to synthesize the model into an ASIP and the software running on it. The synthesized ASIP corresponds to the class library used in the OO model, and hence, can serve other (and future) applications that use the same class library. This is an important advantage over other ASIP-based synthesis approaches [9], since they merely consider a set of given applications and do not directly involve themselves with future ones.

One key point in ODYSSEY ASIP is the choice of the instruction-set: methods of the class library that is used in the embedded application constitute the ASIP instruction-set. The other key point is that each instruction can be dispatched either to a hardware unit (as any traditional processor) or to a software routine. Consequently, an ASIP instruction is the quantum of hardware-software partitioning, and moreover, it shows that the ASIP internals consists of a traditional processor core (to execute software routines) along with a bunch of hardware units (to implement in-hardware instructions).

3 Polymorphism-Aware Bus Protocol

As mentioned before, The NoC implementation for interconnecting processing units (traditional processors or hardware units) in OO-ASIPs [2] is a considerable approach to avoid the overhead of the implementation of polymorphic method calls. In this approach, an NoC is utilized to interconnect processing units. Method calls are accomplished through packets that invoke processing units. Each packet includes a source address, a destination address, method parameters, and lengths of the parameters. Routing of a packet through the NoC is equivalent to dynamic binding of the called method to processing units. When a method call occurs, depending on the destination address, the packet may reach to different processing units, but it belongs to the one that the destination address designates.

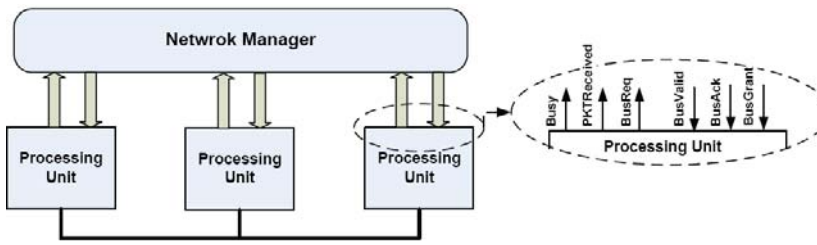


Fig. 1. The proposed configuration of the network for an OO-ASIP with three processing units

In the case of a common bus for interconnecting processing units, we propose Fig. 1 as the network configuration of an OOASIP. In this approach, all processing units transmit packets to each other through a common bus, and their handshaking is performed through the network manager. Each processing unit does handshake with the network manager through six signals:

Busy signal indicates that the processing unit is determining the destination address of the received packet.

PKTReceived signal indicates that the processing unit is the owner of the received packet.

BusReq is a signal that is asserted by the processing unit to request the bus.

BusValid is a common signal to indicate that there is a packet on the bus.

BusAck is a signal from the network manager to indicate that the receiver has received the packet successfully.

BusGrant signal determines the grant of the bus requester.

3.1 The Network Protocol of the Packet Transmitter

A processing unit asserts its BusReq signal when it wants to put a packet on the bus. By means of asserting the related BusGrant of the bus requester, the network manager passes control of the bus to the requester based on the arbitrating algorithm which we present later. The requester puts its packet when it finds out that its BusGrant is active. In this case, two situations may take place.

The first case occurs when the owner of the packet receives the packet properly. The network manager makes the transmitter aware of this situation by asserting the BusAck of the transmitter. Then, the transmitter deactivates its BusGrant and releases the bus when it realizes that its BusAck is asserted and the current transaction accomplishes successfully. The second case occurs when the owner of the packet is not in receiving state and is executing a method. In this case, the network manager announces the situation to the transmitter by keeping the related BusGrant of the transmitter inactive. Hereafter, the transmitter releases the bus, and keeps asserting its BusReq until it can transmit its packet.

Therefore, a packet transmitter has no handshake with receivers directly. This property of the protocol allows the transmitter to transmit its packet without concerning about the state of the packet owner and the network.

3.2 The Network Protocol of the Packet Receiver

Being in receiving state, the processing units deal with each packet once at a transaction. Therefore, the processing units, which are in receiving state, can be assumed as packet receivers, but certainly not as packet owners. If there is a packet transmitter, the network manager will assert the BusValid signal that is a common signal. Thereby, receivers activate their Busy signal. Hereafter, two situations may occur for a packet receiver. First, the packet receiver is the owner of the packet. In this case, after dealing with the packet, the packet owner activates its PKTReceived signal and then deactivates its Busy signal to indicate this situation to the network manager. Second, the packet receiver is not the owner of the packet. After dealing with the packet, the packet receiver deactivates its Busy signal to indicate that it is not the packet owner. If this situation occurs for all packet receivers, it will be a sign for the network manager that the packet owner is not in receiving state and it should abort the current transaction and then start another one.

3.2 The Network Manager

The network manager includes the arbiter that follows an arbitrating algorithm, and the network controller which generates the necessary signals to accomplish the protocol.

Our proposed algorithm for the arbiter is a mixture of round-robin and priority algorithms. The proposed algorithm gives each processing unit a chance based on the round-robin algorithm to transmit its packet through the bus. If the processing unit given a chance has asserted its BusReq, the transaction will be started by the network controller. Otherwise, the arbiter arbitrates between bus requesters based on a simple priority algorithm. Since bus requesters have the same chance to access the bus, the starvation problem never occurs. In addition, we can assign a priority to each processing unit in accessing the bus. In general, if the following statement is true:

$$P_1 > P_2 > \cdots > P_n > 0 . \tag{1}$$

Where, P_i is the priority of unit processing i such that $i \in \mathbb{N}$ and $i \le n$, then the following statement will be realized:

$$(n \times T_{RR}) > W_n > \cdots > W_2 > W_1 > 0 . \tag{2}$$

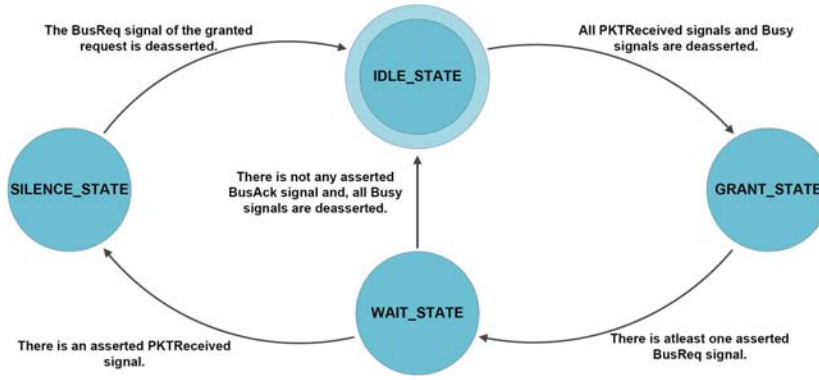


Fig. 2. The state machine of the network controller

Where, W_i is the average waiting time of processing unit *i* to access the bus, and T_{RR} is the round-robin time.

Once the arbiter determines which requester can access the bus, the network controller starts the transaction and generates the necessary signals based on the protocol. The network controller utilizes a state machine that has four states as depicted in Fig. 2.

Idle state indicates that there is no packet on the bus, and the BusValid signal is not asserted. The network controller will stay in this state as long as there is at least one asserted Busy or BusAck signal. This means that this state assures that the packet receivers get ready for the next transaction. In this state, BusAck, BusGrant, and BusValid of all processing units are deactivated by the network controller. Grant state denotes a new transaction. The network controller will stay in this state as long as there is no one to request the bus. In this state, network manager asserts the BusValid and the BusGrant of the bus requester that the arbiter has specified and then goes to the next state. Wait state allows the packet receivers to deal with the packet of the granted transmitter. The network controller will stay in this state as long as there is at least one asserted Busy signal, and it does not find out any activated PKTReceived. In this state, if the network manager finds out an activated PKTReceived, it will assert the BusAck that belongs to the granted transmitter, and then it will go to the next state. Silence state denotes that there is a packet owner. In other words, the network controller enters this state when the packet of the granted transmitter is received successfully by the packet owner. The network controller will stay in this state as long as the granted transmitter keeps its BusReq signal active.

In this protocol, a packet transmitter calling a method transmits its packet on the bus without doing direct handshake with the packet owner, and it accompanies bus transactions independent of the state of the packet owner. These features are what the polymorphism scheme and dynamic binding of a method to a processing unit demand at runtime. Fig. 3 shows an accomplished transaction.

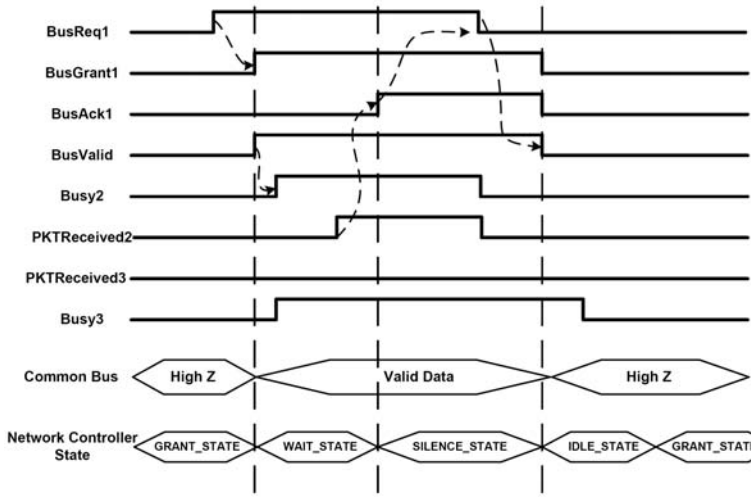


Fig. 3. A typical transaction with three processing units when accomplished

4 Experimental Results

Here, we illustrate the appropriateness of our solution by a simple program that multiplies two matrices in floating point representation, and has a watchdog timer to detect deadlocks. The application is implemented in two variants: the first variant calls methods sequentially with a simple network manager, while the second one exploits the proposed protocol and calls methods concurrently. The target OO-ASIPs consist of three hardware units (Add, Multiply and WD timer) along with a traditional processor. Table 1 shows the synthesis results of the target OO-ASIPs for a typical technology. As Table 1 shows, our proposed protocol rarely enforces extra overhead to hardware units and imposes little overhead to the network manager. Moreover, our solution provides the capability of concurrent execution of processing units as well as polymorphism in calls.

Table 1. The synthesis outcome of a typical application

Variant	Area (gates)				Runtime
	Add	Multiply	WD timer	Network Manager	(clock ticks)
Sequential with a simple network manager	1386	1299	885	170	6092
Concurrent with the proposed protocol	1395	1268	897	182	4865

5 Related Work

There are few projects or methodologies of embedded system design based on object-oriented modeling. Here we briefly describe the ones which are likely related to our work.

The OASE project [10] is an approach based on source-to-source transformation to translate object-oriented models to synthesizable Hardware Description languages (HDLs). For virtual methods, which are involved in polymorphic calls, OASE generates a switch-case statement to bind the appropriate method implementation at run time. Certainly, this generation for each call requires extra hardware-overhead. Therefore, there is no common bus through which methods are invoked, and consequently, there is no need for an arbiter.

The ODETTE project [11] at the University of Oldenburg, Germany, considers attributes of objects as a state machine where the contents of state machine determine the value of attributes. It conceives methods of an object as an event to transit between states. When a virtual method is called, all the candidate method implementations are invoked and then the next state is determined through a mechanism that decides which output of invoked methods should be applied. Therefore, ODETTE cannot call two virtual methods of the same object concurrently; as a result, it does not need an arbiter to invoke virtual methods similar to the OO-ASIP.

For both of the above works, polymorphism capability is achieved by paying more hardware or less concurrency in invoking virtual methods.

6 Conclusions

In this paper, we proposed a protocol for the network of an OO-ASIP that prevents contention on the bus. The processing units do handshake with each other through a unit called network manager that consists of an arbiter and a state machine to control the bus transactions. The proposed arbiter exploits such a modified variant of round-robin algorithm that it never leads to the starvation problem. The protocol does not violate polymorphic concurrent calls in an OO-ASIP and has little overhead. Moreover, it can be used in an OO-ASIP with asynchronous processing units.

References

1. Goudarzi, M., Hessabi, S., Mycroft, A.: Object-Oriented Embedded System Development Based on Synthesis and Reuse of OO-ASIPs. Journal of Universal Computer Science 10(9), 1123–1155 (2004)
2. Goudarzi, M., Hessabi, S., Mycroft, A.: Object-Oriented ASIP Design and Synthesis. In: Forum on specification and Design Languages (FDL 2003), Frankfurt, Germany (2003)
3. Goudarzi, M., Hessabi, S., Mycroft, A.: Overhead-free Polymorphism in Network-on-Chip Implementation of Object-Oriented Models. In: Design, automation and test in Europe (DATE 2004), Washington, DC, USA, p. 21380. IEEE Computer Society, Los Alamitos (2004)
4. IBM Corporation: CoreConnect bus architecture (2007), `http://www-03.ibm.com/chips/products/coreconnect`

5. ARM Corporation: AMBA 2 Specification (2007), http://www.arm.com/products/solutions/AMBA_Spec.html
6. Pyoun, C.H., Lin, C.H., Kim, H.S., Chong, J.W.: The Efficient Bus Arbitration Scheme in SoC Environment. In: The 3rd IEEE International Workshop on System-on-Chip for Real-Time Applications, pp. 311–315 (2003)
7. International Technology Roadmap for Semiconductors (2003), http://public.itrs.net
8. Keutzer, K., Malikand, S., Newton, A.: From asic to asip: The Next Design Discontinuity. In: The 2002 IEEE International Conference on Computer Design: VLSI in Computers and Processors (ICCD 2002), Washington, DC, USA, p. 84. IEEE Computer Society, Los Alamitos (2002)
9. Benini, L., Micheli, G.D.: Networks on Chips: a New soc Paradigm. IEEE Computer 35(1), 70–78 (2002)
10. OASE Project: Objektorientierter hArdware/Software Entwurf (2004), http://www-ti.informatik.uni-tuebingen.de/~oase/
11. ODETTE Project: Object-oriented co-DEsign and functional Test (2003), http://odette.offis.de

Efficient VLSI Layout of WK-Recursive and WK-Pyramid Interconnection Networks

Saeedeh Bakhshi[1,2] and Hamid Sarbazi-Azad[1,2]

[1] Sharif University of Technology, Tehran, Iran
[2] IPM School of Computer Science, Tehran, Iran
{bakhshi,azad}@ipm.ir

Abstract. The WK-recursive mesh and WK-pyramid networks are recursively-defined hierarchical interconnection networks with excellent properties which well idealize them as alternatives for mesh and traditional pyramid interconnection topologies. They have received much attention due to their favorable attributes such as small diameter, large connectivity, and high degree of scalability and expandability. In this paper, we deal with packagibility and layout area of these networks. These properties are of great importance in the implementation of interconnection networks on chips. We show that WK-recursive, mesh-pyramid and WK-pyramid networks can be laid out in an area of O(N2) which is the optimal area for VLSI layout of networks. Also, we obtained the number of tracks for laying nodes in a collinear model, where the chips are multi-layered.

Keywords: VLSI layout, Thompson grid model, Multilayer grid model, WK-recursive, WK-pyramid, Interconnection networks.

1 Introduction

The WK-recursive network which was originally proposed to connect massively parallel computers in traditional interconnection network architectures [1], is a class of recursively scalable networks that can be constructed hierarchically by grouping basic modules. Any *d*-node complete graph, can serve as the basic modules. The $WK(d,t)$ is a network of *t* levels whose basic modules are K_d. Previous works relating to $WK(d,t)$ topology have shown that this network has the advantages of high scalability as a result of its recursive structure, small diameter, small average inter-node distance, and small network cost and a robust topology.

In section 2 we formally state some definitions, which are used through the paper, collinear layouts and their special cases are introduced in this section. This section also introduces two models of VLSI computation, denoted as Thompson model and Multilayer model; We used these model throughout this paper. Layout of WK-Recursive network is proposed in section 3. In section 4 we computed an efficient area complexity for a layout of WK-Pyramid network. Finally, the paper is concluded in section 5.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 123–129, 2008.

2 Definitions and Preliminaries

It is natural to model interconnection networks with graphs that have nodes representing processing units and communication units (switch) connected with edges representing data streams between the nodes. The WK-recursive network [1] is a class of recursively scalable networks denoted as $WK(d,t)$ that is constructed by hierarchically grouping basic modules each a *d*-node complete graph, K_d. $WK(d,t)$ consists of d $WK(d,t-1)$'s each of them could be considered as a super-node. These $WK(d,t-1)$'s are connected as a complete graph.

Definition 1. A t-level WK-recursive network $WK(d,t)$ with amplitude d and expansion level t, consists of a set of nodes $V(WK(d,t)) = \{a_t a_{t-1}...a_1 \mid 0 \le a_i < t\}$. The node with address schema $A = (a_t a_{t-1}...a_1)$ is connected to 1) all the nodes with addresses $(a_t a_{t-1}...a_1 k)$ that $0 \le k < t, k \ne a_1$, as sister nodes and 2) to node $\left(a_t a_{t-1}...a_{j+1} a_{j-1} (a_j)^j\right)$ if for one *j*, $1 \le j < t$; $a_{j-1} = a_{j-2} = ... = a_1$ and $a_j \ne a_{j-1}$, as a cousin node. Notation $(a_j)^j$ denotes *j* consecutive a_j's. We name the nodes with address schema (1^n), (2^n) … and (t^n) as extern nodes. For $1 \le l \le t$, there are exactly W^{t-l} different $WK(d,t)$ sub-networks in a $WK(d,t)$ could be numbered 1 to *d*. Fig.1 shows the structure of the $WK(4,2)$.

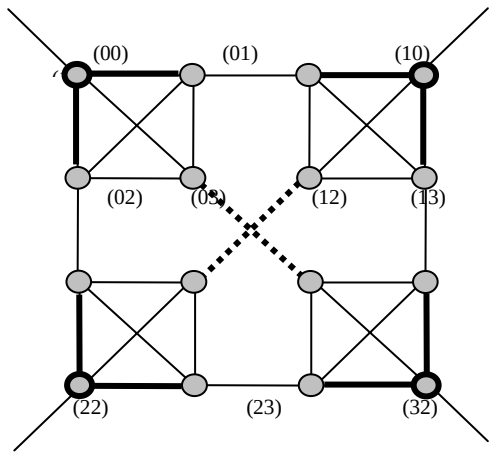


Fig. 1. The topologies shape of a $WK_{(4,2)}$ network. Extern nodes are shown with bold lines.

Definition 2. A *Wk-pyramid* of *n* levels, denoted by $P_{WK,n}$, consists of the set of nodes $V(P_{WK,n}) = \{(k, a_k a_{k-1}...a_1) \mid 0 \le k \le n, 0 \le a_i < 4\}$. A node $V = (k, a_k a_{k-1}...a_1)$ is said to be a node at level *k*. All the nodes in level *k* form a $2^k \times 2^k$ WK-recursive network. We refer to the adjacent nodes at the same level as *sister* nodes. The node *V* is also

connected to nodes (k+1, $a_k a_{k-1}\ldots a_1 0$), (k+1, $a_k a_{k-1}\ldots a_1 1$), (k+1, $a_k a_{k-1}\ldots a_1 2$), and (k+1, $a_k a_{k-1}\ldots a_1 3$).), for $0 \le k < n$, in level k+1, as the *children* nodes. Also, it is connected to node (k-1, $\left\lfloor \frac{x}{2} \right\rfloor, \left\lfloor \frac{y}{2} \right\rfloor$), in level k-1, as the *father* node. The apex node is (0,0) which can be alternatively denoted by $P_{WK,n}$▲. The degree of the apex is 4. The addresses of extern nodes are (n,0^n), (n,1^n), (n,2^n) and (n,3^n) where a^n denotes n consecutive a's; these may alternatively be shown as $P_{WK,n}$◤, $P_{WK,n}$◥, $P_{WK,n}$◣ and $P_{WK,n}$◢, respectively. Each node in the n-th layer, has exactly 5 neighbours while the node degree of nodes in the middle layers is 9. It can be easily shown that there are a total of $N = \sum_{k=0}^{n} 4^k = (4^{n+1}-1)/3$ nodes in a $P_{WK,n}$ and the diameter of $P_{WK,n}$ is $D = 2n-1 \approx \log N$. In addition, the $P_{WK,n}$ is Hamiltonian connected. A network is Hamiltonian connected if we can make a Hamiltonian path starting from any node in the network and ending at any other node. Fig. 2 shows a $PWK(2)$.

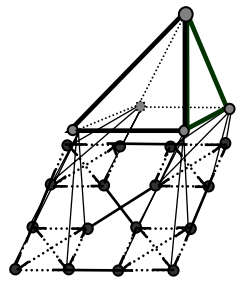


Fig. 2. The topologies shape of a *PWK (2)* network

Definition 3. A VLSI layout is called collinear if all the nodes are placed along a straight line.

2.1 Thompson Grid Model

Thompson proposed a mathematical model for VLSI computations, which is widely accepted, and is known as the Thompson grid model [10]. In this model, he presumed the chip is consisting of some vertical and horizontal tracks which are spaced apart at unit intervals. Two layers of interconnect are used to route the wires. Vertical wires are routed in one layer while horizontal wires are routed in the other. The circuit is viewed as a graph G in which vertices correspond to processing elements and edges to wires.

The graph is then embedded in a two-dimensional grid. An embedding of a graph G in a Thompson grid is an assignment of nodes of G to intersection points in the grid and the edges of G to paths along the grid tracks. The area of a layout is the area of the smallest upright rectangle that contains all the nodes and wires. When there are two layers of wires, it is guaranteed that we can lay out the network within the area. The maximum wire length is the length of the longest wire in the layout.

2.2 The Multilayer Grid Model

In the multilayer grid model, a network is viewed as a graph whose nodes correspond to processing elements and edges correspond to wires. The nodes and edges of the graph are then embedded in a 3-D grid, where edges have unit width, can run along grid lines, but cannot cross or overlap with each other (i.e., the paths for embedding these edges must be edge- and node-disjoint). The area *A* of a layout is defined as the area of the smallest upright rectangle along the *x*-*y* directions that contains all the nodes and wires. The volume of a layout is equal to the number *L* of layers times its area *A*.

In the multilayer 2-D grid model, the nodes of the graph are embedded in the 2-D grid of the first layer (i.e., $z = 1$).

In the multilayer 3-D grid model, the nodes of the graph are embedded in L_A layers of the 3-D grid. These L_A layers are called "active layers" and do not need to be consecutive layers. The motivations for using multilayer layout models include the significant reduction achieved in the layout area, volume, and maximum wire length required, leading to considerable improvements in both hardware cost and performance [7].

3 Layout of WK-Recursive

In this section, we use the Thompson grid model and the multilayer model to find an efficient layout for WK-Recursive network. First we obtain the area and number of tracks needed for packaging WK-Recursive on a one layer chip, then by using these layouts, we investigate on S-layer layout of network.

3.1 2-D Layout of WK-Recursive

An *WK(d, t)* contains *d* subgraphs isomorphic to *WK(d, t-1)*, each pair of which are connected by one link. If we let $WK(d,t-1)$ as a supernode (like the way Yeh and Parhami did in [7, 8, 9], then the *WK(d, t)* becomes a *d*-supernode complete graph.

To lay out a *WK(d, t)*, we first place nodes belonging to each *WK(d, t-1)* into a block which we call *(t-1)*-block, and lay out the *WK(d, t)* using that of complete graph, stated in [9]. Then we should continue this process recursively until all the graphs are laid out.

Theorem 1. A *WK(d, t)* can be laid out in $N^2/16+o(N^2)$, where $N=d^t$.

Proof. To lay out *WK(d, t)*, based on the layout of the complete graph [9], the area of $N^2/16+o(N^2)$ will be required. For level h-block we first connect the wires outside the block to appropriate *(h-1)*-blocks within it, and then follow the technique of laying out the K_{C^h} graph. Hence, the layout area will be obtained from:

$$\frac{d^{2t}}{16}+o\left(d^{2t}\right)+\frac{d^{2(t-1)}}{16}+o\left(d^{2(t-1)}\right)+\ldots+1=\frac{d^{2t}}{16}+o\left(d^{2t}\right)$$

And the theorem follows. Fig. 3 shows a collinear layout of $WK(3,2)$.

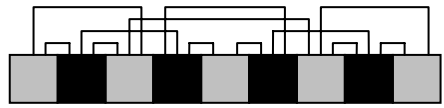


Fig. 3. A collinear layout of $WK(3,2)$

3.2 Multilayer Layout of WK-Recursive

The idea of laying out the interconnection networks in a 2-D or 3-D grid models first proposed in [7]. Yeh, Varvarigos and Parhami showed that many of the known networks can be laid out more efficient using this idea. Here we use their results to obtain the layout area, volume and maximum wire length of *WK(d, t)*.

In order to obtain the multilayer layout of *WK(d, t)* we use the same bottom-up approach, used in [7] for *k-ary n-cube* and several other networks.

Theorem 2. A *WK(d, t)* needs $\frac{d^{t+1}-d^t+2d^{t-1}-2}{d-1}$ tracks to layout collinearly.

Proof. We start with the *d*-node complete graph. A collinear layout of a complete graph can be obtained by placing the *d* nodes along a row and connecting nodes, like the complete graph layout [7]. Let the number of tracks which are needed by $WK(d,t-1)$ be $f_d(t-1)$. To obtain the collinear layout of *WK(d, t)* we need *d* copies of $WK(d,t-1)$ which are placed in a space, horizontally *d* times of the space of $WK(d,t-1)$ and two extra tracks to connect the level L inter-block links.

Therefore, the number of tracks needed to layout the *WK(d, t)* is $f_d(t)=df_d(t-1)+2$, and because $f_d(1)=d$, we have

$$f_d(t)=df_d(t-1)+2=d^t+2\left(d^{t-2}+d^{t-3}+...+d+1\right)$$

$$f_d(t)=d^t+\frac{2\left(d^{t-1}-1\right)}{d-1}=\frac{d^{t+1}-d^t+2d^{t-1}-2}{d-1}$$

Now, we use the approach of orthogonal multilayer layout [7] scheme to obtain an *S*-layer layout.

The number of tracks per layer above a row is $\left\lceil \frac{d^{\lfloor (t-1)/2 \rfloor}\left(d^2-d+2\right)-2}{\lfloor S/2 \rfloor (d-1)} \right\rceil$ and the number of tracks per layer to the right of the column is $\left\lceil \frac{d^{\lceil (t-1)/2 \rceil}\left(d^2-d+2\right)-2}{\lfloor S/2 \rfloor (d-1)} \right\rceil$. So the area of the S-layer $WK(d,t)$ is: $\frac{2N^2}{\lfloor S^2/2 \rfloor d}+o\left(\frac{N^2}{S^2 d}\right)$

And its volume is: $\frac{2N^2}{\lfloor S/2 \rfloor d}+o\left(\frac{N^2}{Sd}\right)$.

4 Layout of WK-Pyramid

In this section we investigate on the VLSI layout of WK-Pyramid network. For simplicity, we first propose a theorem for obtaining number of tracks needed to lay out mesh-pyramid network, collinearly. The next theorem, theorem 4, specifies the area complexity of mesh-pyramid network.

Theorem 3. A P_n needs $\frac{3N}{2}+O\left(\sqrt{N}\right)$ tracks to layout collinearly, where N is the number of nodes of P_n.

Proof. To find the collinear layout complexity of P_n we use a recursive process. Let the number of tracks which are needed by P_n be $T\left(P_n\right)$. For P_0 we have $T\left(P_0\right)=1$, because there is one isolated node that can be put in an square.

P_1 includes a $M_{2^1\times 2^1}$ and a P_0, connected to all of the mesh vertices. So $T\left(P_1\right)=4N\left(M_{2^0\times 2^0}\right)+T\left(M_{2^1\times 2^1}\right)$, in which $N\left(M_{2^i\times 2^i}\right)$ is the number of nodes in $M_{2^i\times 2^i}$, i.e. $N\left(M_{2^i\times 2^i}\right)=2^i\times 2^i=2^{2i}$.

Since $T\left(M_{2^i\times 2^i}\right)=O\left(2^i\right)$, we have $T\left(P_1\right)=4+O\left(1\right)=O\left(1\right)$.

For the P_n, similarly we need to lay P_{n-1} out and add a $M_{2^n\times 2^n}$ with $4N\left(M_{2^{n-1}\times 2^{n-1}}\right)$ edges connected the mesh to the last layer of P_{n-1}. Hence the collinear layout of P_n needs $T\left(P_n\right)=4N\left(M_{2^{n-1}\times 2^{n-1}}\right)+T\left(M_{2^n\times 2^n}\right)=2^{2n}+O\left(2^n\right)$ tracks. Since $N=N\left(P_n\right)=\frac{\left(4^{n+1}-1\right)}{3}$, we have $T\left(P_n\right)=2^{2n}+O\left(2^n\right)=\frac{3N}{2}+O\left(\sqrt{N}\right)$ and the theorem follows.

Theorem 4. A P_n can be laid out in $\frac{27}{2}N^2+O\left(N\sqrt{N}\right)$ area.

Proof. Since mesh-pyramid has a maximum vertex degree of 9, if we let length of any node be 9, i.e. each node in the layout consumes a 9 square area, and then the theorem follows, clearly.

Now, we use the two previous theorems to obtain the number of tracks in the collinear layout of WK-pyramid.

Theorem 5. A $PWK\left(t\right)$ needs $\frac{13}{6}N-\frac{2}{3}$ tracks to layout collinearly, where N is the number of nodes of $PWK\left(t\right)$.

Proof. The proof is similar to that of mesh-pyramid. For the $PWK\left(t\right)$, it is enough to lay out $WK\left(4,t\right)$, and then connect each node belongs to the last layer of $PWK\left(t\right)$ to the corresponding node in $WK\left(4,t\right)$. So the collinear layout of $PWK\left(t\right)$ needs $T\left(PWK\left(t\right)\right)=4N\left(WK\left(4,t-1\right)\right)+T\left(WK\left(4,t\right)\right)$ tracks. Since $N\left(WK\left(4,t-1\right)\right)=4^{t-1}$

and by theorem 2, $T\left(WK\left(4,t\right)\right)=\frac{4^{t+1}-4^{t}+24^{t-1}-2}{3}=\frac{7}{6}\times 4^{t}-\frac{2}{3}$, we have $T\left(PWK\left(t\right)\right)=4^{t}+\frac{7}{6}\times 4^{t}-\frac{2}{3}=\frac{13}{6}\times 4^{t}-\frac{2}{3}$ and the theorem follows.

Theorem 6. A $PWK\left(t\right)$ can be laid out in $\frac{39}{2}N^{2}+O\left(N\right)$ area.

Proof. Again the same as mesh-pyramid, Since $PWK\left(t\right)$ has a maximum vertex degree of 9, the theorem follows.

5 Conclusion

WK-recursive mesh and WK-pyramid are defined recently as a successor of mesh and traditional pyramid network. They possess many desirable properties like a small diameter, small network cost, and high degree of expandability. Packagibility is an important factor in implementation of interconnection networks. In this paper, we showed how to embed these networks on chips with little area cost and we obtained efficient layouts for their VLSI implementation. These areas are optimum and shows that Wk-recursive and WK-pyramid networks can be put on chips of a little area of $O\left(N^{2}\right)$.

References

1. Della Vecchia, G., Sanges, C.: A Recursively Scalable Network VLSI Implementation. Future Generation Computer Systems 4(3), 235–243 (1988)
2. Leighton, F.T.: Introduction to parallel algorithms and architectures: Arrays, Trees and Hypercubes. Morgan Kaufmann, San Mateo (1992)
3. Bondy, J.A., Murty, U.S.R.: Graph Theory with Applications. North-Holland, NY (1976)
4. Hoseiny Farahabady, M., Sarbazi-Azad, H.: The WK-Recursive Pyramid: An Efficient Network Topology. In: 8th International Symposium on Parallel Architectures, Algorithms and Networks ISPAN 2005, pp. 312–317 (2005)
5. Hoseiny Farahabady, M., Sarbazi-Azad, H.: The grid-pyramid: A generalized pyramid network. Journal of Supercomputing 37, 23–45 (2006)
6. Chen, G.H., Duh., D.R.: Topological Properties, Communication, and Computation on WK-Recursive Networks. Networks 24, 303–317 (1994)
7. Yeh, C.-H., Varvarigos, E.A., Parhami, B.: Multilayer VLSI layout for interconnection networks. In: Proc. Int'l Conf. Parallel Processing, pp. 33–40 (2000)
8. Yeh, C.-H., Parhami, B.: On the VLSI Area and Bisection Width of Star Graphs and Hierarchical Cubic Networks. IEEE Trans. Computer (2001)
9. Yeh, C.-H., Parhami, B., Varvarigos, E.A.: The recursive grid layout scheme for VLSI layout of hierarchical networks. In: Proc. Merged Int'l Parallel Processing Symp. & Symp. Parallel and Distributed Processing, pp. 441–445 (1999)
10. Thompson, C.D.: A Complexity Theory for VLSI. PHD thesis, Carnegie-Mellon Univ. (1980)

Energy Adaptive Cluster-Head Selection for Wireless Sensor Networks Using Center of Energy Mass

Ehsan Akhtarkavan[1] and Mohammad Taghi Manzuri Shalmani[2]

[1] Islamic Azad University, Arak, Iran
akhtarkavan@ispa.ir
[2] Sharif University of Technology, Tehran, Iran
Manzuri@sharif.edu

Abstract. A set of small battery-operated sensors with low-power transceivers that can automatically form a network and collect some desired physical characteristics of the environment is called a wireless sensor network. The communications must be designed to conserve the limited energy resources of the sensors [14].By clustering sensors we can save energy. In this paper, we introduce a new concept called "Center of Energy Mass" which is a combination of both energy level and location of the nodes which is used to form the new factor of "distance of the nodes to the CEM ".Distance of the nodes to the CEM is used together with Probability Density Function of the normal distribution in optimizing LEACH's cluster head selection algorithm. We optimized LEACH's random Cluster-Heads selection algorithm by means of finding the CEM, to ensure balanced energy depletion over the whole network thus prolonging the network lifetime. Simulation results show that our algorithm improves First Node Dies by 23.5% and Half Nodes Die by 5.6%.

Keywords: Wireless Sensor Network, LEACH, energy adaptive, center of energy mass, normal distribution, CEMLeach.

1 Introduction

A wireless sensor network is composed of a large number of sensor nodes and a base station that serves as a gateway to some other networks. Sensor nodes sense their environment, collect sensed data and transmit to the base station. However, they are limited in power, computational capacity and memory. Moreover, they have only short-range radio transmission. [13]

Note that these nodes are resource-constrained devices and last until their battery has been depleted. When half of the nodes die, the network is gone. Due to these peculiar characteristics of the sensor nodes, the traditional routing protocols employed in wired networks and ad hoc networks are not suitable for the sensor networks [13].

The cost of transmitting a bit is higher than a computation [3] and hence it may be advantageous to organize the sensors into clusters. In the clustered environment, the data gathered by the sensors is communicated to the Base Station through Cluster-Heads [15]. Since the sensors are now communicating data over smaller distances in the clustered environment, the energy spent in the network will be much lower than

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 130–137, 2008.

the energy spent when every sensor communicates directly to the Base Station. Hence cluster formation with optimal Cluster-Heads selection can drastically affect the network's communication energy dissipation [15]. The LEACH protocol present in [1] is the first cluster-based routing protocol for wireless sensor network and has motivated the design of many other protocols [6], [5], [4] which follow a similar concept.

In this paper, we present a modification of LEACH's Cluster-Heads selection algorithm to further reduce and balance the total energy dissipation of sensors. The rest of the paper is organized as: Following the introduction, section 2 introduces some preliminary notions concerning the proposed protocol. The widely used hierarchical protocol called LEACH [1] is explained in section 3. Section 4 presents our CEM (center of energy mass) approach which is called CEMLeach. Simulation results of CEMLeach are presented in section 5. Section 6 discusses possible future research directions and concludes this study.

2 Preliminaries

2.1 Radio Energy Dissipation Model

We use the free space radio model as stated in [7], [2] with E_{elec}=50 nJ/bit as energy being dissipated to run the transmitter or receiver circuitry and ε_{amp}=10 PJ/bit/m^2 as the energy dissipation of the transmission amplifier to achieve an acceptable E_b/N_o (Fig. 1).

A d^2 energy loss is used due to channel transmission. Thus, to transmit an l-bit message to a distance d, the radio expends:

$$\begin{aligned} E_{Tx}(l,d) &= lE_{elec} + l\varepsilon_{amp}d^2 \\ E_{Rx}(l,d) &= lE_{elec} \end{aligned} \tag{1}$$

With l as the length of the transmitted / received message in bits, d as the distance between transmitter and receiver nodes [7].

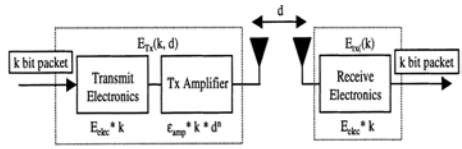


Fig. 1. Radio energy dissipation model [2]

3 LEACH

3.1 LEACH Protocol Architecture

Low-Energy Adaptive Clustering Hierarchy (LEACH) presented in [1], [2] provides an elegant hierarchical protocol that uses localized coordination to enable scalability and robustness for micro sensor networks, and exploits data aggregation in the routing

protocol to reduce the amount of data packets that must be transmitted to the Base Station. LEACH divides the operation of the entire network into many rounds. Each round consists of a set-up phase and some number of time frames that construct the steady state phase.

When clusters are being formed, each node *n* autonomously decides if it will be a cluster head for the next round or not. The selection is stochastically: Each node determines a random number between 0 and 1. If this number is lower than a threshold $T_1(n)$, the nodes becomes a cluster head. $T_1(n)$ is determined according to the equation 2:

$$T_1(n) = \frac{p}{1 - p * (r \bmod \frac{1}{p})} \tag{2}$$

For nodes that have not been Cluster-Heads in the last 1/*P* rounds, otherwise $T_1(n)$ is zero. Here *P* is an a priori determined number that determines the average number of Cluster-Heads during a round, *r* is the number of the current round.

4 The Proposed Clustering Algorithm

To prolong the life time of the network first we must define it. The definition of the lifetime of a micro sensor network is determined by the kind of service it provides [10]. Hence, three new approaches of defining lifetime are proposed: First Node Dies (FND), half of the Nodes Dies (HND) and Last Node Dies (LND). [10] For a cluster-based algorithm like LEACH the metric LND is not important since more than one node is necessary to perform the clustering algorithm. Hence, in this paper we just discuss these two metrics, FND and HND.

A first approach to increase the lifetime of a clustered sensor network which uses LEACH is to incorporate the remaining energy level available in each node in calculation of the threshold. In our algorithm we introduce a new measure to incorporate the energy level of the nodes and the center of energy mass into the calculation of the threshold based on equation 1. In fact considering only one factor, like energy, is not suitable to select the Cluster-Heads properly. This is because other conditions like centrality of the nodes with respect to the entire cluster, also gives a measure of the energy dissipation during transmission for all nodes. The more central the node is to a cluster, the more is the energy efficiency for other nodes to transmit through that selected node. The concentration of the nodes in a given region also affects in some way for proper Cluster-Heads selection. It is more reasonable to select a Cluster-Heads in a region, where the node concentration is high .Center of energy mass is a place where the aggregate energy of the network is uniformly distributed around it. Center of energy mass is calculated as:

$$X_{cem} = \frac{\sum x_i E_i}{\sum E_i} \ , \ Y_{cem} = \frac{\sum y_i E_i}{\sum E_i} \tag{3}$$

There is a trade-off between the node's remaining energy level ($E_{current}(n)/E_{total}$) and distance of the node from the CEM. According to definition of the CEM, it is a good

idea to elect nodes which are far from the CEM, because it makes dissipation of energy more uniform. In addition we have to elect nodes with higher remaining energy level to prolong the life time of the network. Thus we multiply these two factors and form a new factor:

$$d_{toCEM} * \frac{E_{current}(n)}{E_{total}(r)}. \tag{4}$$

To normalize this factor we subtract the reversed form of equation 4 from one, and consequently the new threshold factor can be determined as:

$$T_2(n) = T_1(n) * (1 - \frac{1}{d_{toCEM} * \frac{E_{current}(n)}{E_{total}(r)}}). \tag{5}$$

Here, d_{toCEM} is the distance of the node *n* from the CEM, $E_{current}(n)$ is remaining energy of node *n*, and $E_{total}(r)$ is the aggregated remaining energy of the network. In this way, d_{toCEM} would be directly proportional to the threshold and it helps nodes far from the CEM to have more chance to be selected as the Cluster-Head. But as shown in section 5 it is not a fair selection for those nodes that are near the CEM and their chance of being a Cluster-Head is drastically decreased. So we have to further optimize equation 5 by incorporating a new factor. Our new factor is derived from the probability density function of the normal distribution [12]. Probability Density Function of the normal distribution is a Gaussian function which has the maximum value in the mean. To increase the probability for nodes which are near the CEM we can use the reciprocal of the Gaussian function and introduce the new threshold factor $T_3(n)$ as follows:

$$T_3(n) = T_2(n) * \exp(\frac{(d_{ave} - d_{toCEM})^2}{2 * v}), \tag{6}$$

Where, d_{ave} is the average-distance of nodes from the CEM and *v* is the variance of nodes' distances from d_{ave}.

It is worthwhile to say that, by $T_2(n)$ factor, we improve the probability of being selected as a Cluster-Head for nodes that are far from the CEM, while the probability of being a Cluster-Head for those that are near the CEM is decreased. But by incorporating the $T_3(n)$ factor we increase the probability of being a Cluster-Head for nodes that are either far from the CEM or next to the CEM, compensating the $T_2(n)$ influence on the probability of nodes that are next to the CEM. Note that, the proposed modification does not change the probability for nodes located within the average distant of nodes from the CEM.

5 Experimental Results

We have implemented the algorithms described in Sections 3 and 4 with OMNet++ 3.3v discrete event simulator [9].In this simulation model we have simulated a

wireless sensor network with 100 sensor nodes which are randomly distributed among a 1000m*1000m area, and the Base Station in located at (x=500,y=1750). We have used the same radio model and, whenever possible, the same parameter settings as Heinzelman et al. [1].In our simulation we consider FND and HND measures to evaluate our algorithms. In addition we recorded values of above mentioned factors to prove our interpretations. Distance of the node to the CEM , d_{toCEM} is the new factor which we proposed but its better to use it together with energy level of the node which is determined as the ratio of current energy of the node to the aggregate energy of the network in the current round ($E_{current}(n)/E_{total}$).As we multiply these two factors we get a value very greater than one and though its reciprocal would fall between 0 and 1.But as we expected the distance of the node to the CEM must directly be proportional to probability of being a Cluster-Head, so we subtracted it from 1 and got the equation 7:

$$Y_1 = (1 - \frac{1}{d_{toCEM}(n) * \frac{E_{current}(n)}{E_{total}}}). \tag{7}$$

In Fig. 2 we have plotted the value of factor Y_1 during every round as a function of distance to the CEM.

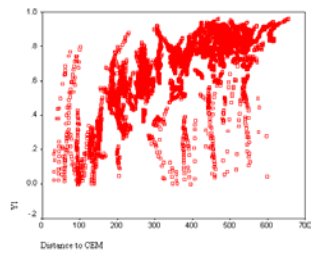


Fig. 2. The value of Y1 vs. distance

As depicted in Fig. 2, in general the value of Y_1 increases as the distance of the node to the CEM is increased, but it has different effects on nodes with different distances to the CEM. It increases the probability of being a Cluster-Head for nodes that their distance to the CEM is larger than the average-distance of nodes to the CEM, and less decreases the probability of being a Cluster-Head for nodes nearer than the average distance. Now to increase the probability of being a Cluster-Heads for node that are next to the CEM we introduce a new factor. We can use the reciprocal of the Gaussian factor and have a new factor Y_2 to have the minimum value in d_{ave} :

$$Y_2 = \exp(\frac{(d_{toCEM}(n) - d_{ave})^2}{2v}). \tag{8}$$

Now, we can plot the value of Y_2 factor as a function of distance to the CEM, as shown in Fig. 2:

Table 1. Quantiles of nodes and each average

N	9581	
Quantile	Percentiles	Average
First	25	280.0947
Second	50	418.9971
Third	75	497.8186

Table 2. FND and HND when using

Protocol	T2(n)		T_3(n)	
	HND	FND	HND	FND
DeterministicLEACH	104	62	104	64
LEACH	106	64	106	62
CEMLeach	108	64	112	79

Fig. 2 demonstrates that Y_2 factor has the minimum value at the average distance point and it increases when going apart in either directions, but it sharply increases when getting near the CEM, thus compensates the shortage of Y_1 factor. According to the distance to the CEM, our algorithm partitions the nodes into three quantiles, first quantile consists of nodes next to the CEM, second quantile consists of nodes within the average distance, and the third quantile consists of nodes with far apart the CEM. In each group we can calculate the averages individually. Results are presented in Table 2.

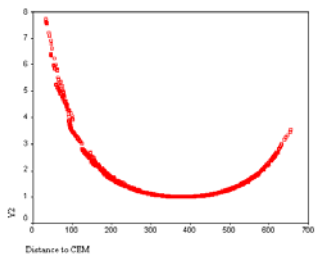


Fig. 3. Y2 vs. distance

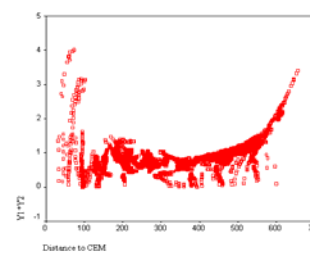


Fig. 4. Y1*Y2 vs. distanc

We hoped to be able to compensate the shortcomings of the Y_1 factor using the Y_2 factor. We multiplied these two factors to incorporate them as well. Next we plot the multiplication of both factors Y_1*Y_2 as a function of distance of the nodes to the CEM. Fig. 3 demonstrates that we increase the probability of being Cluster-Heads of nodes that are either next to the CEM or very far from the CEM. By these modifications we ensure that nodes residing in the mentioned groups form clusters within their own groups. In addition it increases the probability of being a Cluster-Heads for nodes which are located either very far from the CEM or next to the CEM. The

probabilities of nodes that are located within the average-distance to the CEM are not affected very much.

Now we take a look at the important measures FND and HND of the network. We have implemented LEACH and Deterministic LEACH protocols in addition to compare the results of our algorithm.

Table 3 shows that using equation 5 ($T_2(n)$) we could not optimize the FND measure of the algorithm at all. In addition it is demonstrated that using $T_2(n)$ we improve HND 2 rounds. Incorporating Y_1 factor and using $T_2(n)$ we could increase HND but could not increase FND.Now we incorporate Y_2 factor and though use $T_3(n)$. In table 3 it is denoted that using $T_3(n)$ we improve FND by 23.4% and according to table.3 we can say that our algorithm optimizes the LEACH's HND measure by 5.66%.According to the results we have increased FND by 23.5% and HND by 5.6%.At last to interpret the situation and to say what's going on, we take a look at the total number of selected Cluster-Heads during simulations of each protocol.

Table 3. Number of Cluster-Heads

Protocol	#rounds	#CH	Mean
CEMLeach $T_3(n)$	112	403	3.59
CEMLeach $T_2(n)$	108	419	3.88
LEACH	106	420	3.96
DeterministicLEACH	104	502	4.82

Table 3. demonstrates that the average number of selected Cluster-Heads in each round has been decreased for our algorithm and it shows that better formation of clusters has resulted in a better performance.

6 Conclusions and Future Work

In this paper, we have presented CEMLeach, an energy-efficient clustering hierarchy protocol which is a modified version of the LEACH. Our simulations have shown that making LEACH aware of its Center of Energy Mass (CEM) increases the lifetime of the wireless sensor network substantially in typical scenarios. We have used the CEM to optimize the formation of clusters and hence increased the probability of being a Cluster-Head for nodes either next to the CEM or very far from the CEM.

References

1. Heinzelman, W., Chandrakasan, A., Bal Krishnan, H.: Energy-Efficient Communication Protocol for Wireless Microsensor Networks. In: International Conference on System Sciences, Hawaii (January 2000)
2. Heinzelman, W., Chandrakasan, A., Balakrishnan, H.: An application Specific Protocol Architecture for Wireless Microsensor Networks. IEEE Transactions on Wireless communications 1(4) (October 2002)

3. Pottie, G.J., Kaiser, W.J.: Wireless Integrated Network Sensors. Communications of the ACM 43(5), 51–58 (2000)
4. Manjeshwar, A., Agarwal, D.P.: TEEN: a routing protocol for enhanced efficiency in wireless sensor networks. In: 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing (April 2001)
5. Manjeshwar, A., Agarwal, D.P.: APTEEN: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks. In: Parallel and Distributed Processing Symposium. Proceedings International, IPDPS 2002, pp. 195–202 (2002)
6. Lindsey, S., Raghavendra, C.: PEGASIS: Power-Efficient Gathering in Sensor Information Systems. In: IEEE Aerospace Conference Proceedings, vol. 3(9-16), pp. 1125–1130 (2002)
7. Rappaport, T.: Wireless Communications: Principles & Practice. Prentice-Hall, Englewood Cliffs (1996)
8. Voigt, T., Ritter, H., Schiller, J.: Utilizing Solar Power in Wireless Sensor Networks. In: IEEE Conference on Local Computer Networks, LCN 2003, Bonn/Königswinter, Germany (October 2003)
9. Varga, A.: The OMNeT++ Discrete Event Simulation System. In: European Simulation Multiconference, Prague, Czech Republic (June 2001)
10. Handy, M.J., Haase, M., Timmermann, D.: Low energy adaptive clustering hierarchy with deterministic Cluster-Heads selection. In: Proc. 4th International Workshop on Mobile and Wireless Communications Network, September 2002, pp. 368–372 (2002)
11. Voigt, T., Dunkels, A., Alonso, J., Ritter, H., Schiller, J.: Solar-aware clustering in wireless sensor networks. In: Proc. Ninth International Symposium on Computers and Communications, June 2004, pp. 238–243 (2004)
12. Mood, A.M., Graybill, F.A., Boes, D.C.: Introduction to the Theory of Statistics, 3rd edn. McGraw-Hill Companies, New York (1974)
13. Tillapart, P., Thammarojsakul, S., Thumthawatworn, T., Santiprabhob, P.: An Approach to Hybrid Clustering and Routing in Wireless Sensor Networks
14. Bandyopadhyay, S., Coyle, E.J.: An Energy Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks. In: IEEE INFOCOM 2003 (2003)
15. Ying, L., Haibin, Y.: Energy Adaptive Cluster-Head Selection for Wireless Sensor Networks. In: Sixth International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT 2005) (2005)

SHRP: A New Routing Protocol to Wireless Sensor Networks

Cláudia J. Barenco Abbas[1], Nelson Cárdenas[1], Giácomo Lobalsamo[2], and Néstor Davila[2]

[1] Universidad Simón Bolívar (USB), Departamento de Computación y Tecnología de la Información, Caracas, Venezuela
[2] Universidad Central de Venezuela (UCV), Escuela de Computación, Facultad de Ciencias, Caracas, Venezuela
ncardenas@ldc.usb.ve, barenco@ldc.usb.ve, globalsamo@gamil.com, nestordavila@gmail.com

Abstract. This paper presents a new routing protocol to Wireless Sensor Networks called SHRP. This protocol has as principal goal the saving of energy and also provides reliability of data delivery. It is a novel proposal as it acts as a proactive protocol in respect to the choosing of routes, taking in consideration just nodes that can contribute to extend the network lifetime, that means, good battery status and good link quality. It also contributes to network management advising a central point about a possible disconnection of a node caused by low battery or large interference periods. Data messages can be added during the forwarding task and redundant sensing data are not sent, so it contributes again with the energy saving. We show that SHRP has a low time of convergence, which means that reacts quickly in case of topology changes, which is common in Wireless Sensor Networks, as the battery lifetime is low and the environment of wireless network uses to suffer from interferences and obstacles.

Keywords: Wireless Sensor Network, Routing Protocol, IEEE 802.15.4.

1 Introduction

Monitoring and controlling activities in different systems are fundamental tasks to improve the systems performance. Advances in micro-electronics have allowed, on a large scale, the integration of sensor, microcontroller and communication components in small footprint and low cost devices. These features are making possible the establishment of monitor and control loop in places were it was not possible or affordable until recent time.

The sensor node could measure data from physical system and sent it, usually via radio transmitter, to a command center or sink node, either directly or through a number of communication and data concentration devices (or gateways). The size and cost reduction of sensors devices has increased the interest about using sets of disposable and unattended sensors. Such situation has motivated intensive research in the recent years, addressing the potential of collaboration among sensors in data collecting, processing and sensing activities management.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 138–146, 2008.

The process of transmitting data to a base station in an energy efficient way is the main goal of WSN (W*ireless Sensor Networks*), as these networks are oriented in extending the batteries life in order to reduce network maintenance costs and to have a well connected topology. In WSN, there is not a traditional protocol as there is in TCP/IP architecture, mainly due to its huge limitation on storage capacity, processing power and battery life, so there have been recent works that intend to propose new protocols [1].

2 Related Works

There are different requirements for each WSN, it depends on the physical environment that is being instrumented and the network characteristics. Some routing protocols for WSN classified by the network architecture are mentioned in this report.

2.1 Flat Networks

In flat networks all network nodes play the same role. The simpler flat protocols are based on flooding and gossip. Flooding strategy is based on sending messages in any direction in order to reach any network node, even its own destiny. This strategy is very simple to implement however it involves sending a lot of messages with a large power consuming, which is not too convenient in WSN. Gossip is base on sending messages, not to all nodes, but to a random subset of neighbors, which reduce the consuming power, but does not have a real warranty that the message is going to arrive to its destiny with an efficient power consuming cost. Others flat proposes have been studied and detailed in [14] like SPIN[2], Directed Diffusion[3] and AIMRP[4].

Flat routing algorithm such as SPIN (*Sensor Protocol Information Negotiation*) and direct diffusion strategies, seems not having many application in wireless network sensors of small sizes, and short distances. In a simple network structure in which are not needed many hops to communicate information between sensor nodes, make no sense to use a protocol that make negotiations of complex routes of many hops.

2.2 Hierarchical Networks

In hierarchical architectures, subsets of nodes called Cluster Heads, have special tasks related to scalability and efficient communications. Proposes like LEACH [5] (*Low-Energy Adaptive Clustering Hierarchy*), TEEN [6] (*Threshold sensitive Energy Efficient sensor Network*) and DIRq [7] suggest a distribute cluster formation with different strategies to contribute to overall system lifetime and energy efficient.

We are interested in considering a protocol that could deal with three different aspects: battery available, number of hops and link quality to guarantee the arrival of messages in the sink node in an energy saving way. We could not find any routing protocol to WSN that use these three parameters together are at the same time be concerned about energy saving and reliability features. By all this reason we propose a new routing protocol.

3 SHRP: Functional Description

The main goal of SHRP protocol is the reliability of data delivery, maintaining the network topology and choosing the best route based on battery lifetime, number of hops and network link quality. SHRP was designed for WSN applications where energy saving is an important issue. As data transmission is the most expensive energy task [8], not all periodic data messages are sent. Two energy-saving policies are implemented by SHRP: (i) sensing data is sent only if its value has changed from the last time; (ii) coordinator nodes can aggregate various data message and send just one message. If the sensing data does not change in a certain period, the data should be sent to detect connection problems.

SHRP protocol monitors the battery available and link quality between neighbor nodes, cutting off nodes from the routing table that can not contribute to maintain a well connected topology.

The link layer of 802.15.4 offers two metrics to measure the link quality: LQI and RSSI. Newer radios that are based on IEEE 802.15.4 standard such as CC2420 implement a LQI (*Link Quality Indicator*) which is believed to be a better indicator than RSSI (R*eceived Signal Strength Indicator)* [2]. *P*rotocol designers are looking for inexpensive and agile link estimators that may choose RSSI over LQI [9], but RSSI at the edge of the threshold of -87 dBm [10] does not have a good correlation with PRR (*Packet Reception Rate*), so SHRP uses both of the metrics to choose the best route.

SHRP uses minimum thresholds to cut off 'bad' neighbor nodes from the routing table. Bad link quality can be caused, for example, by interference, multipath or path loss. All of these problems are reflected in LQI and RSSI metrics, so SHRP cuts off nodes when there is bad link quality and automatically insert the node when problems do not exist more.

SHRP protocol also cuts off nodes that do not have enough battery available to execute what we call "the minimum task cycle" [11] (see Equation 1).

$$\text{Minimum Task Cycle} = \text{CCA} + \text{Sensing Task} + \text{Transmission task} + \text{Reception task} + \text{Idle Period task} \quad (1)$$

To evaluate different routes to the sink node SHRP analyze the number of hops to sink node, trying to pass through the minor number of hops, so saving energy. To maintain the network topology, SHRP chooses the route that has more energy available among all the possible routes until the sink node. Each node must decide the next hop to reach to sink node using local information. To do this, each node should know the minimum value of each metric (i.e. battery available, link quality, etc) for each possible route defined in the table route.

3.1 Topology Configuration

SHRP is a hierarchical and proactive routing protocol. It has three different types of nodes: sensor nodes (SN), coordination nodes(CN) primary and secondary and sink node.

SHRP is intended to be used in a sensor network that has a static topology. The static topology can be defined after a site survey process which makes some warranty about network physical connection as any SN should communicate with at least one

CN. Nevertheless, every SN should be associated with only one CN and use a configuration protocol to learn who its CN node [15] is.

Each CN can have one or more neighbors, some of them can be reached directly from Sink node. Each sent message will arrive into a Sink node through this CN that should be part of the best route defined by SHRP protocol. During network deployment some politics could be taken in order to guarantee components redundancies. Even different nodes could be connected to the same sensors (if sensing device is expensive or is difficult to find). If we can aggregate additional node to establish redundant routes, this could impact network topology, increasing network lifetime. This is feasible due to the low cost of motes devices.

Redundant CNs can be used as secondary coordinators (CNsec) that must not be far from its associated primary. Each CNsec could be in sleep mode and periodically it should check if its primary CN is alive. If it does no receive any response from CNsec during a certain time, it assumes that its primary CN has gone, so has to become a primary CN.

3.2 SHRP Messages

We describe two different types of messages: data and control. Control messages are used to define the tree routes of the SHRP. NMI, Hello message and Alarms are control messages that we will explain in detail below. Data messages transmit monitoring information to the Sink node.

Network Information Message (NMI). The purpose of this message is to provide local information to each node in the network. All metrics in NMI are collected, calculated and transmitted in a peer to peer fashion between neighbor nodes. The NMI format is detailed in figure 1.

Sink node must send one NMI message every NMI_INTERVAL time, to detect changes in the network topology. The message information includes the current link quality between the node and the address of the sender node. NMI is sent in a broadcast way. When receiving the message the node should check whether the sender is a 'good' neighbor. With the information received in each message, the node must recalculate all metrics and update the information about the 1-hop neighbor. Before sending its NMI message, the CN node makes an auto evaluation of its battery available metric (battRem). If the value is below a threshold, it does not send the NMI message to its neighbor, as it is not a 'good' neighbor to others nodes. If a node decides not to send its NMI message, it must send an special alert message called "*Control Alarm Message*" (CAM) to the sink node, to inform that it will leave the list of neighbors soon. The main idea is to avoid loose the connectivity in the network. In an intuitive form we can say that a 'good' neighbor is a CN node that has enough battery to guarantee message delivery and offer a reliable link, in terms of radio signal, to reach the Sink node.

Hello Message. Sending a NMI message is not a very frequent process mainly due that network convergence requires enough time to NMI message reach all nodes in the network. However, changes in the network caused by battery problems, interferences, etc can appear in anytime. Once NMI has defined the network topology, SHRP uses Hello message (figure 2) to detect network changes. This message is cheaper, in

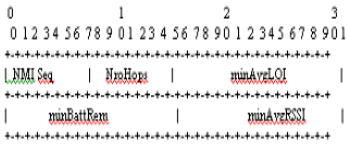


Fig. 1. NMI Message format

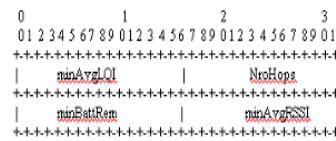


Fig. 2. Hello Message format

terms of overhead, than NMI because each node send periodically just one hello message to its neighbors. Each CN must send a Hello message each HELLO_INTERVAL interval. When a node receives a hello message, it should update the neighbor metrics in its neighbor table and guarantee that uses the best route between the neighbor options. It's possible that the node have better route to sink (i.e. due to some node in a previous stage died) so the node can change their route to sink. Next hello message will report this change to their neighbors propagating in this way topological changes. If CN does not receive hello message from node A, then must verify if node A is part of the best route to sink and whether there is other routes to reach it. In this case, CN changes its best route, otherwise CN must send an alarm called "Orphan node" to sink in order to send a NMI message and reconfigure the whole network.

3.3 SHRP Metrics

SHRP use the following metrics: battery available, link quality and number of hops to select the best route into the sink node. LQI and RSSI are used as link quality indicators.[14]. Due to metrics like LQI can suddenly change [9] SHRP protocol uses an average for these metrics calculated within a window time, so it rapidly changes and do not affect SHRP decisions.

Route selection can not be based only on the data of the next step, since it can lead us to select a route that offers a lower delivery guarantees in later stage. So we propose that some metrics used by the protocol, would be calculated considering information from every node that will participate in the selected route. Keep information for every node in the route is not feasibly due to store restrictions in the node. Every node calculates for each route, the minimum value for each metric, using NMI information. The minimum value of the route until the previous step is included in the NMI payload, to update it just needed to add local metrics value. For instance, in the battery available metric (battRem), each node must calculate the lowest battery available for every possible route to reach Sink node and choose the greatest (we called this result MaxMinBattRem). Idem for RSSI and AvgLQI metrics.

To avoid loops in the route selection, is not necessary to considering number of hops as first metric, because using the duple: maxMin metric value and the id of the node that generated it, SHRP can difference two distinct routes. If two routes have the same duple value and different number of hops, we are sure that it is not a good route (can be a loop) so SHRP should discard it.

4 Experimental Results

As previously shown, in order to establish a routing logical topology with SHRP protocol, the NC nodes have to maintain its routing tables with data from the neighbors. The sink node has to send a NMI message by flooding. When NC nodes receive all NMI messages, update its routes and calculate the best route from itself to the sink node.

(1) (2) (3) (4)

Fig. 3. Topologies of the experiments

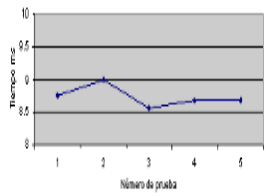


Fig. 4. Convergence Time of topology 1 – SHRP

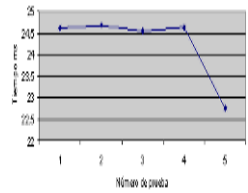


Fig. 5. Convergence Time of topology 2 – SHRP

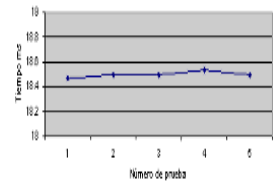


Fig. 6. Convergence Time of topology 3 – SHRP

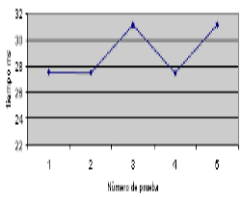


Fig. 7. Convergence Time of topology 4 – SHRP

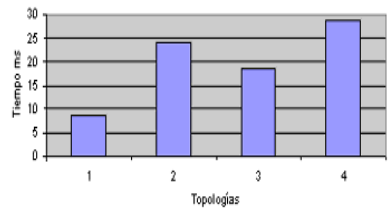


Fig. 8. Comparison of Convergence Times - SHRP

To do some experimental tests we have defined the ‘convergence time’ as a period that takes long for all NC nodes update its routing tables, so it includes all transmission and processing times. In a Wireless Sensor Network, the clocks that come with each mote works independently, so in order to calculate the convergence time of SHRP protocol, we have implemented a synchronization mechanism based on the TPSN scheme [16]. The experimental tests were done in four different topologies (see figure 3) to observe the worst case (maximum) of convergence time. We repeated five

times each experiment. Tests were done using motes TMote from Moteiv Company with TinyOS operating system and CC2420 Radio Chip [17].

In the star topology (see figure 3.1) all NC nodes have a direct link with the Sink Node. The results of the convergence time of this topology can be seen on figure 4. In the second topology the NC1 and NC2 nodes have a direct link with the Sink Node, but node NC3 just has a direct link with NC1 and NC2 nodes. With this topology NC3 has a redundant path until sink node and has two 'father' nodes (see figure 5).

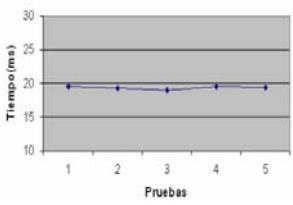


Fig. 9. Convergence Time of topology 1 – HTS

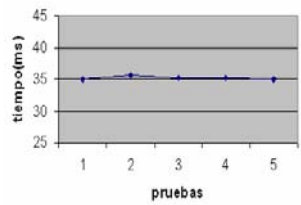


Fig. 10. Convergence Time of topology 2 – HTS

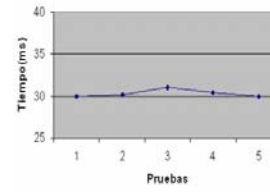


Fig. 11. Convergence Time of topology 3 – HTS

As we can notice in figure 6, topology 3, NC2 and NC3 are two hops far from the Sink Node, having just one 'father' node (NC1). In the fourth topology (see figure 7) there are three hops until Sink Node with the following direct links: Sink-NC1, NC1-NC2 and NC2-NC3, showing a tandem topology.

The consolidation of the data is shown in figure 8 with the media of the results from each topology. As we can see, topology 1, 3 and 4 maintain the same behavior of the convergence time, increasing in a lineal way regarding to the number of hops. We obtained this result due to each NC node has only one neighbor to learn route table. In the second topology, the convergence time had a different behavior when comparing it with the topology 3. Both of them are two hops from the Sink Node; however the processing time is greater in topology 2 because NC3 node in topology 3 has more neighbors, so NC3 node has to process more NMI messages and calculate more in order to have the best route until Sink Node.

With these experiments we can conclude that SHRP protocol has a media convergence time of 8 ms for each direct link that has to exchange NMI messages, and this time increases linearly with the increase of neighbor nodes. The second experiments that we have done are the comparison of SHRP protocol and HTS (*Hop to Sink Protocol*) protocol [18]. HTS protocol is a simple routing protocol that we have proposed and implemented. The only metrics that it uses are the number of hops and the Sink Node has to receive all routing table from all nodes to decide the best route and send the decision to each node. HTS protocol looks like a link state protocol but just Sink Node knows all routes from all NC nodes.In this study we have the same topologies shown in figure 3 with the same parameters previously described.

As we can see in figure 9, the Convergence Time for the first topology reached 20 ms. With SHRP we had a maximum convergence time of 9 ms. Now in topology 2 (figure 10), the maximum Convergence Time was 35 ms, meanwhile with SHRP protocol it was 24.7 ms. In figure 11 is shown the Convergence Time of HTS in

topology 3. As we can see the maximum Convergence Time was 31 ms and with SHRP protocol it was 18,5 ms.

These experiments shows that SHRP protocol has a better convergence time in respect to a simple protocol as HTS, that just chooses the neighbor based on number of hops until Sink Node. Link state Protocols like HTS, where each node has to know all the routes to Sink Node, can have a large convergence time, being a problem when there is a topology that changes constantly as in Wireless Sensor Network, as we have low battery time, interference and obstacles problems. So we can conclude that protocols that just need to have neighbor information to decide the best route shows to have a lower convergence time, being more interesting to WSN.

5 Conclusion

SHRP protocol uses battery available, quality of link and distance, in number of hops, until the sink node in order to choose the best routes. These metrics together contribute to have a reliable and energy aware routing protocol. It is a novel proposal as it uses a mixing of metrics in the choosing of the best route and also works in a proactive manner. In the establishing of the link quality metric it not only uses the LQI indicator, but also RSSI indicator, what is a novel usage in the area of routing protocol.

Experimental results shows that SHRP has a low Convergence Time in respect to link state protocols, like HTS, where each node has to know all the topology in order to decide the best route. Convergence Time is an important aspect as Wireless Sensor Networks uses to have an instable topology due to low battery lifetime and obstacles, and interference problems, so the update of routing tables has to be a common task.

References

1. Al-Karaki, J.N., Kamal, A.E.: Routing Techniques in Wireless Sensor Network: a Survey" Wireless Communications. IEEE 11(6), 6–28 (2004)
2. Kulik, J., Heinzelman, W., Balakrishnan, H.: Adaptive Protocols for Information Dissemination in Wireless Sensor Networks. In: International Conference on Mobile Computing and Networking. Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking, Seattle, Washington
3. Intanagonwiwat, C., Govindan, R., Estrin, D.: Directed diffusion: A scalable and robust communication paradigm for sensor networks. In: 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCOM 2000), Boston, MA, USA, August 2000, pp. 56–67 (2000)
4. Kulkarni, S., Iyer, A., Rosenberg, C.: An Address-light, Integrated MAC and Routing Protocol for Wireless Sensor Networks, April 26 (2005)
5. Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H.: Energy-Efficient Communication Protocol for Wireless Microsensor Networks. In: Proc. Of Hawaiian International Conference On Systems Science (January 2000)
6. Manjeshwar, A., Agrawal, D.: TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks. In: WIRELESS (April 2001)

7. Chatterjea, S., De Luigi, S., Havinga, P.: DirQ: A Directed Query Dissemination Scheme for Wireless Sensor Networks. In: WSN 2006 Wireless Sensor Networks (July 2006)
8. Polastre, J., Hui, J., Levis, P., Zhao, J., Culler, D., Shenker, S., Stoica, I.: A unifying link abstraction for wireless sensor networks. In: Proceedings of the Third ACM Conference on Embedded Networked Sensor Systems (SenSys 2005) (2005)
9. Shan, L., Zhang, J., et al.: ATPC: Adaptive Transmission Power Control for Wireless Sensor Networks. In: SenSys 2006, Bolder, Colorado, USA (November 2006)
10. Srinivasan, K., Levis, P.: RSSI is under appreciated. In: Proceedings of the Third ACM Workshop on Embedded Networked Sensors (EmNets 2006) (May 2006)
11. Polastre, J., Hill, J., Culler, D.: Versatile low power media access for wireless sensor networks. In: Second ACM Conference on Embedded Networked Sensor Systems (2004)
12. CC2420 datasheet, http://www.chipcon.com/files/CC2420_Data_Sheet_1_4.pdf
13. Barenco, C., Gonzalez, R., Cardenas, N.: Proposta de um protocolo de roteamento para Redes Sem Fio de Sensores em Poços de Petróleo. In: I2TS 2006 - 5th International Information and Telecommunication Technologies Symposium, Cuiaba, Brasil (December 2006)
14. Barenco, C., Gonzalez, R., Cardenas, N.: A proposal of a Wireless Sensor Network Routing Protocol. In: PWC 2007 – Personal Wireless Communication, Prague, Czech Republic, September 2007. IFIP, vol. 245, pp. 410–417. Springer, Heidelberg (2007)
15. González, R., Cárdenas, N., Barenco, C.: Un protocolo de enrutamiento con características de tolerancia a fallos y autoconfiguración en Redes Inalámbricas de Sensores. In: CLEI2007 XXXIII Conferencia Latinoamericana de Informática, San José, Costa Rica (October 2007)
16. Chen, S., Dunkels, A., Österlind, F., et al.: Time Synchronization for Predictable and Secure Data Collection in Wireless Sensor Networks. In: The Sixth Annual Mediterranean Ad Hoc Networking Workshop, Corfu, Greece, June 12-15 (2007)
17. http://www.moteiv.com
18. Implementación de un Protocolo Simple basado en número de saltos al sink para comunicación en una Red Inalámbrica de Sensores. Technical Report. Cristiam da Silva (October 2007)

Improvement of MAC Performance for Wireless Sensor Networks

M.H. Fotouhi Ghazvini, M. Vahabi, M.F.A. Rasid, and R.S.A. Raja Abdullah

Department of Computer and Communication Systems Engineering, Faculty of Engineering, Universiti Putra Malaysia, Selangor, Malaysia
fotouhi@ieee.org, fadlee@eng.upm.edu.my

Abstract. The fast progress of research on energy efficiency in wireless sensor networks, and the need to compare with the solutions adopted in the standards motivates the need for this work. In the analysis presented, the star network configuration of 802.15.4 standard at 868 MHz is considered for a Zigbee network. In this paper, we analyze the active duration of the superframe and entered the sleep mode status inside this period. It happens when sensors do not have any data to send. The nonpersistent CSMA uses the adaptive backoff exponent. This method helps the network to be reliable under traffic changes due to save the energy consumption. The introduction of sleep state has shown incredible reduction of the power consumption in all network load changes.

Keywords: Zigbee, IEEE 802.15.4, Media Access Control, Wireless Sensor Network, Energy Efficiency.

1 Introduction

Use of this technology appears to be limited only by our imagination and ingenuity. A diverse set of applications for sensor networks encompassing different fields have already emerged including medicine, agriculture, environment, military, inventory monitoring, intrusion detection, motion tracking, machine malfunction, toys and many others. In the medical field sensor networks can be used to remotely and unobtrusively monitor physiological parameters of patients such as heartbeat or blood pressure, and report to the hospital when some parameters are altered [1, 2, 3, 4].

Energy efficiency is a fundamental theme pervading the design of communication protocols developed for wireless sensor networks (WSNs), including routing and MAC layer protocols. Some papers focus on this field of study just in network layer by techniques like clustering and hierarchies [11,13] or compare this performance with delay as other major performance metric in WSN [12,14]. Two main mechanisms for achieving low energy consumption in energy-constrained WSNs in MAC layer are how to deal with collision and how to reduce idle listening period in the superframe duration. In this approach, each sensor node cycles between awake state and sleep state. The period of superframe duration consists of these two modes due to the nodes contribution in network traffic. The backoff exponent which is a factor of defining the time waiting for channel access is dynamically changed due to load

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 147–152, 2008.

changes. This paper focuses on the design of a special MAC by looking into both critical issues related to energy consumption, which are collision and idle listening.

2 Sleep State and Adaptive Backoff Exponent Algorithm

The IEEE 802.15.4 achieves a low-duty cycle operation by means of its beacon-enabled mode. In this mode, a PAN coordinator periodically disseminates a superframe structure bounded by a beacon frame into the network and manages its active/inactive period. Any associated devices are allowed to communicate in the active period and conserve energy by turning off their transceiver during the inactive period. However, when we assume to stay in the active mode, the idle listening limits the overall performance of the system due to consuming more power.

In our scheme, a coordinator can observe the network traffic due to the data information associated with devices. It can manage the PANs to go to sleep mode when they don't have packet to send. In this model, devices having no data traffic are not required to continuously maintain an active state even they are in their superframe duration part. If a node finds pending data traffic in its queue buffer, it tries to convey the traffic information to its coordinator.

In MAC 802.15.4, there is an exponential increase in the number of packet drops at high data rates. During channel access the CSMA-CA algorithm is allowed to use only a very small range of backoff exponents (macMinBE to aMaxBE), where the minimum BE, a device can support, is indicated by macMinBE=3 and the maximum by aMaxBE=5. Since the variable BE determines the number of backoff slots the devices shall wait to access the channel, the higher the value of BE, the longer the device will spend trying to access the channel in some cases. The longer wait adds up to the power consumption of the device. Therefore, a lower BE range is desired which will ensure that the devices will never spend too much time waiting for channel access, which it is not even sure of. This often allows two or more devices ending up using the same number of backoff slots. As a result they detect an idle channel simultaneously and proceed with their transmissions which results in frequent collisions.

The ABE algorithm [7] is based on providing higher range of backoff exponents to the devices, to reduce the probability of devices choosing the same number of backoff slots to sense the channel. The minimum backoff exponent is also variable; hence devices are not likely to start off with the same backoff exponent when they wish to start a data transmission.

3 Assumptions and Proposed Model

We confine our evaluation to 802.15.4 networks operating in a one-hop star topology which is preferred for applications such as WBANs and Zigbee, where the coordinator is a device like a PDA, a cell phone, a bed-side monitoring station or a Zigbee sensor device that collects data. Such star topologies may also exists inside clusters in large networks of 802.15.4 devices. We consider that all sensors are in a beacon-enabled mode. In this scheme, 15 nodes are associated with a common coordinator in

a one-hop star topology in a center of a circle, which the others are in the peripheral of the network area, and all nodes are within the carrier sensing range of each other. This ensures that an ongoing transmission will not be disrupted by other nodes.

We assume that the superframe works just in active mode, this will be achieved by taking same values for SFO and BCO. Although inactive period allows nodes to sleep periodically and conserve energy, but as it introduces undesirable delays in delay-critical monitoring applications like WBANs, particularly at higher beacon orders, this part has been eliminated. We concentrate our analysis on the uplink mode (communication from nodes to coordinator), this allows nodes to enter the sleep state depending on their own availability of data to transmit rather than having to stay awake for the entire active period and compensate the elimination of inactive period.

The MAC IEEE 802.15.4 has active and inactive mode. The inactive period allows nodes to sleep periodically, but most of the applications are delay-critical, so we just select the active part.

It is apparent that it takes time for each state transition, but as mentioned in [8] this time is very short and doesn't affect the performance of our work and also have a significant effect on the overall energy consumption in the network. As indicated before, we consider a beacon-enabled network with no inactive part in the superframe, in which the nodes can sleep since the power consumption in the awake state is several times more than that might be considered reasonable, it is not sufficient to keep the nodes in the awake state when not transmitting. So we allow nodes to enter the shutdown state when they have not packet to send to the coordinator. We consider the case when radios are allowed to enter the shutdown state if there is no packet to be transmitted. Radio shutdown has been shown to be very effective in conserving nodes' energy consumption.

On the other hand we have used the adaptive backoff exponent [7] to decrease the collision in the network. This idea provides higher range of backoff exponents to the devices. The macMinBE is dynamically incremented or decremented due to load changes and is initialized with 3 in first three beacon intervals. Also the devices are allowed to use a higher backoff exponents (aMaxBE=7). The nodes are grouped into two, the first group nodes are whose macMinBE (minimum backoff exponent) is to be decreased by 1. These are the nodes which contribute less to the network traffic compared to the other nodes. Group two comprises whose macMinBE is to be increased by 1. When the number transmitted packets (a measure of traffic) by group-2 nodes differ from group-1 nodes by at least PKT-DIFFERENCE number of packets. Fig. 1. shows the algorithm of our work in a flow chart. In various states of a radio the power expenditure differs, including long-term average dissipation in the various states as well as power consumption during state transitions. For illustrative purposes, we consider the Chipcon 802.15.4-compliant RF transceiver, ZMD44101. The Chipcon radio supports the following four states:

1. ***Shutdown or Sleep***. The crystal oscillator is switched off and the radio is completely disabled waiting for a startup strobe.
2. ***Idle***. The crystal oscillator is turned on and the radio is ready to receive commands to switch to Transmit or Receive state.
3. ***Transmit***. The radio is actively transmitting.
4. ***Receive.*** The radio is actively receiving.

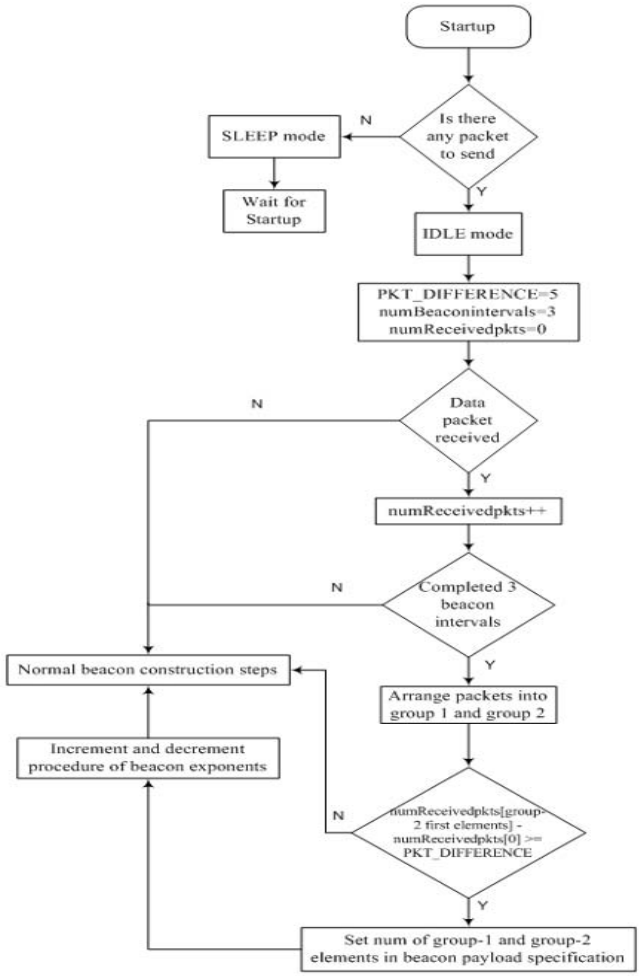


Fig. 1. The proposed model flow chart

It is apparent that it takes some time to switch from one state to another, and this aspect will affect the overall energy consumption in wireless sensor networks, particularly those characterized by low transmission duty cycles. As indicated before, we consider a beacon-enabled network with no inactive part in the superframe in which the nodes can sleep. Since the power consumption in the Idle state is several times more that what might be considered reasonable, it is not sufficient to keep the nodes in the Idle state when not transmitting or reciving. We must therefore find alternative ways to put the nodes to sleep even in the active part of the superframe. However, for benchmarking purposes, we start out by leaving the nodes in Idle state when not active. We allow the nodes to enter the Shutdown state when not active and evaluate its impact on the power consumption, average delay and throughput.

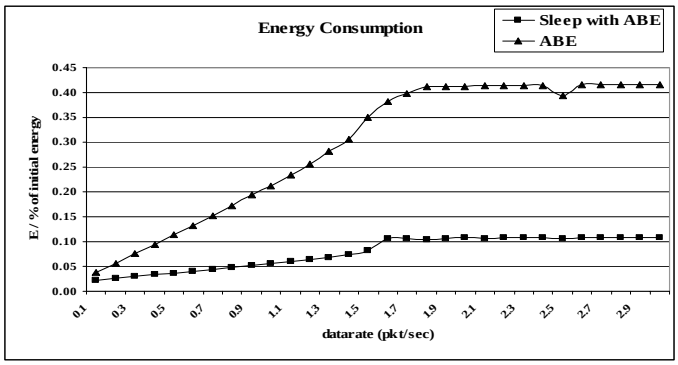


Fig. 2. Energy consumption analysis

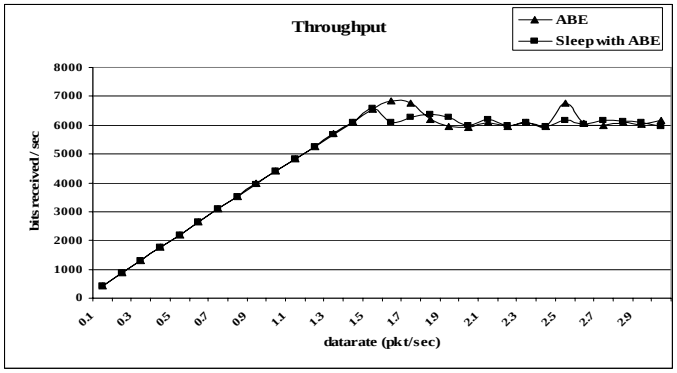


Fig. 3. Throughput analysis

4 Simulation and Results

This simulation is conducted on a star network topology with 15 nodes and 8 traffic flows with CBR traffic and one way communication from the node to the coordinator. Each packet is assumed to be of 70 Bytes. A drop Tail queue is used. Two-Ray ground propagation model with an Omni Antenna is used. The nodes are all placed at a distance of 10m from the coordinator. The results are conducted with ns2 [10] using the software modules provided by [9]. Fig. 2. shows the analysis of energy consumption in terms of data changes from 0-3 kbps. As we have discussed before and the results show, by using sleep period with ABE we get incredible saving of energy compared to pure ABE model. This improvement does not affect the throughput of the system as we can see in Fig. 3., it shows approximately the same results as ABE.

References

1. Malan, D.J., Welsh, M., Smith, M.D.: A public key infrastructure for key distribution in TinyOS based on elliptic curve cryptography. In: Proceedings of the 1st IEEE communications Society Conference on Sensor and Ad-Hoc Communications and Networks (2004)

2. Gao, T., Greenspan, D., Welsh, M.: Improving patient monitoring and tracking in emergency response. In: Proceedings of the International Conference on Information Communication Technologies in Health (2005)
3. Amto, G., Chessa, S., Conforti, F., Macerta, A., Marchesi, C.: Health care monitoring of mobile patients, Ercim news, vol. 60 (2005)
4. Malan, D., fulford-Jones, T., Welsh, M., Moulton, S.: CodeBlue: an ad hoc sensor network infrastructure for emergency medical care. In: Proceedings of the International Workshop on Wearable and Implantable Body Sensor Networks (2004)
5. ZigBee Alliance, ZigBee Specifications, version 1.0 (2005)
6. Institute of Electrical and Electronics Engineers, Inc., IEEE Std. 802.15.4-2003 Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (LR-WPANs). IEEE Press, New York (2003)
7. Parakash Rao, V., Marandin, D.: Adaptive Backoff Exponent Algorithm for Zigbee (IEEE802.15.4). In: Koucheryavy, Y., Harju, J., Iversen, V.B. (eds.) NEW2AN 2006. LNCS, vol. 4003, pp. 501–516. Springer, Heidelberg (2006)
8. Ramachandran, I., Das, A.K., Roy, S.: Analaysis of the contention access period of IEEE 802.15.4 MAC. ACM Transaction on Sensor Networks 3(1), Article 4 (2007)
9. Zigbee Software Modules, City College of NewYork, `http://ees2cy.engr.ccny.cuny.edu/zheng/pub/`
10. The network simulator (Version 2), `http://www.isi.edu/nsnam/ns/`
11. Fotouhi Ghazvini, M.H., Vahabi, M., Rasid, M.F.A., Raja Abdullah, R.S.A.: Optimizing Energy Consumption in Hierarchical Clustering Algorithm for Wireless Sensor Networks. In: Proceedings 2007 IEEE Int'l Conference on Telecommunications and Malaysia Int'l Conference on Communications, ICT-MICC 2007, Malaysia (2007)
12. Vahabi, M., Fotouhi Ghazvini, M.H., Rasid, M.F.A., Raja Abdullah, R.S.A.: Trade-off between Energy Consumption and Target Delay for Wireless Sensor Network. In: Proceedings 2007 IEEE Int'l Conference on Telecommunications and Malaysia Int'l Conference on Communications, ICT-MICC 2007, Malaysia (2007)
13. Rasid, M.F.A., Raja Abdullah, R.S.A., Fotouhi Ghazvini, M.H., Vahabi, M.: Energy Optimization with Multi-level Clustering Algorithm for Wireless Sensor Networks. In: Fourth IEEE and IFIP International Conference on wireless and Optical Communications Networks, WOCN 2007, Singapore (2007)
14. Rasid, M.F.A., Raja Abdullah, A., Kamariah Noordin, N., Vahabi, M., Fotouhi Ghazvini, M.H.: Energy-Aware Data Diffusion Protocol with Reduced Source-to-Sink Delay for Wireless Sensor Network. In: Proceedings Asia Pacific Conference on Communications 2007 APCC 2007, Thailand (2007)

Route Optimization Security in Mobile IPv6 Wireless Networks: A Test-Bed Experience

Abbas Mehdizadeh[1], S. Khatun[1], Borhanuddin M. Ali[2], R.S.A. Raja Abdullah[1], and Gopakumar Kurup[2]

[1] Dept. of Computer & Communication Eng, Faculty of Eng., UPM, Malaysia
[2] MIMOS Berhad, Technology Park Malaysia, 57000 Kuala Lumpur, Malaysia
mehdiizadeh@ieee.org, {sabira,rsa}@eng.upm.edu.my,
{borhan,gopakumar.kurup}@mimos.my

Abstract. Route Optimization (RO) is standard in Mobile IPv6 (MIPv6) to route packets between Mobile Node (MN) and Correspondent Node (CN) using shortest possible path. It provides better bandwidth and faster transmission. RO greatly increases the security risk. In this paper, focus is given on enhanced security scheme in terms of RO based Test-bed evaluation experiment. An enhanced security algorithm is developed on top of MIPv6 RO to secure data. This algorithm is able to detect and prevent the attacker from modifying the data with using an encryption algorithm by cost of little bit increase but tolerable delay. The real-time network Test-bed is implemented to prove the efficiency of proposed method. The experimental results show that the proposed security scheme increases the security performance of the network. This gives advantage of safe communication that can significantly improve the data security of RO while maintaining the quality of other network performance.

Keywords: Mobile IPv6, Route Optimization, Security, IPv6 Test-bed.

1 Introduction

Mobile IPv6 (MIPv6) is an IP-layer protocol for enabling mobility in IPv6 networks on top of the existing IP infrastructure [1]. The MIPv6 allows nodes to be reachable by a static IP address which is called Home Address (HoA). When the Mobile Node (MN) is far away from Home Agent (HA), the packets between MN and Correspondent Node (CN) have to travel via HA. This inefficient routing is called triangle routing. To rectify this problem, MIPv6 introduces a Route Optimization (RO) mechanism. When the MN receives a tunneled packet, it must decide to establish RO. MN send Binding Update (BU) message to CN contains mobile home address and CoA, the CN stores these information in Binding Cache to use to send packet to CoA instead of sending to HoA. Unfortunately BUs can be used by attackers to launch the attack. However MIPv6 uses IPsec to protect signaling between MN and HA, and Return Routability (RR) procedure for protection of the signaling between MN and CN in RO [6], [9], [10], [11]. Even with these security protocols, an attacker on the path between MN and CN, which is called Man-In-The-Middle (MITM) attack, can monitors and modifies the packets payload.

In this paper we proposed a method to detect such attacks, if an attacker is found then an encryption method is used to prevent the attacker to modify the packet data.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 153–159, 2008.

The organized of this paper is as follow: Section 2 presents the background of MIPv6 Security, as well as the review of security threats. In Section 3 the focus is given on the IPsec and RR. The limitations and problems are pointed out in Section 4. The new security method, assumptions, and Test-bed design are presented in Section 5, Followed by result and discussion and finally conclusion of the paper.

2 Security Background of Mobile IPv6

Whenever RO is used it will give attackers a good opportunity to exploit MIPv6 by sending false BUs during the process. If BU were not authenticated at all, an attacker could fabricate and send spoofed BUs from anywhere in the Internet. There is no way of telling which addresses belong to mobile nodes that really could send BUs and which addresses belong to stationary nodes. If the data in the packets is not protected cryptographically, this can lead to compromise of secrecy and integrity [3], [4], [10], [12]. Threats are classified based on the capabilities of an attacker as follows [4]:

- Attacker located anywhere in the Internet:
- In this case, attacker can tamper with the CN Binding Cache (BC) by sending BU to CN and launch Denial-of-Service (DoS) or MITM attack. An attacker can send fake BUs to HA, CN or MN and cause BU flooding or DoS attack.
- Attacker located in the same link/subnet as MN or CN:
- If attacker learn about CN that MN is communicating with and determine to which CN the MN sending BUs, the attacker can send a spoofed BU to CN and MN. This is leading to DoS, MITM or altering the contents of the traffic.
- Attacker located in the same link/subnet as HA:
- Acting as the HA by an attacker is leading to flooding attack, MITM or DoS. An attacker is able to send spoofed BUs/Bas and launch DoS, MITM, changing the route of packets and altering the contents of the traffic.

3 Mobile IPv6 Security Solutions

3.1 IPsec

The IETF has developed the IPsec protocol suite as an extension to the basic IP protocol [2]. It is based on modern cryptographic technologies making possible strong data authentication and encryption. It works on the network level, Layer three on the protocol stack, so it is invisible to applications. This means that any applications running on the network will also benefit. IPsec is compatible with current Internet standards in both IPv4 and IPv6, but in IPv6, IPsec is defined as mandatory feature. The IPsec protocol suite has two modes of operation, Tunnel mode and Transport mode, and includes three IP extensions [5], [7], [8], [13]:

- Authentication Header (AH) provides source authentication, and allows the receiver to verify the identity of the sender, prevents IP Spoofing.
- Encapsulated Security Payload (ESP) provides data encryption and ensures that data has not been read by anyone, prevents Packet Sniffing.

- Internet Key Exchange (IKE) allows two or more parties to agree on authentication methods, encryption algorithms, and securely exchange keys.

3.2 Return Routability Procedure (RR)

To establish Route Optimization, the MN sends a BU to the CN with its current CoA. To prevent attackers from sending false BU, the BU is authenticated using a cryptographic signature that verifies the CN can contact the MN using both addresses is referred to as the Return Routability procedure.

The following steps describe the sequence of event in RR procedure for securing RO [1], [10]. After MN moves to another network and acquires a CoA:

- The MN sends a Care-of Test Init (CoTI) and a Home Test Init (HoTI) to the CN. The CoTI is addressed directly to the CN, and HoTI is routed through the HA.
- The CN sends Care-of Test (CoT) and Home Test (HoT) responses to the MN. The CoT is addressed directly to MN, and HoT is routed through the HA.
- The MN calculates a Binding Management Key (Kbm) from the CoT and HoT.
- The mobile node uses the Kbm to calculate a cryptographic authentication value for BU information, and sends the BU to the CN with the authentication value and index values.
- The CN verifies the BU and sends a BA if MN sets the A-bit in its BU.

4 Limitations and Problems

There is a concern regarding to the performance of IPsec. The required processing power is large for security functions, especially for IPsec. Many users would not have enough throughputs. Even with IPsec, the majority of vulnerabilities are at the application layer, something that IPsec will do nothing to prevent. IPsec is not usable for authentication between MN and CN, because no pre-shared secret key can be used.

The AH and ESP do not provide security against traffic analysis, it is not economical to provide protection against traffic analysis at IP layer. Also AH and ESP do not provide non-repudiation when used with default algorithms.

IPsec tunnels break through firewall or Network Address Translation (NAT) or tunneled IPsec traffic may contains malicious data. Quality-of-Service also does not work in IPsec.

5 The New Security Proposed Method

Our proposed method is based on data security analysis against all above solutions mentioned so far. In this paper we propose a method to detecting attacks against data in route optimization on MIPv6 network, if an attacker is found then using encryption to protect the data. After establishment of RO, MN and CN communicate directly. The attacker is located on the path between MN and CN, and modifies the data sending from MN to CN. When MN is sending packets, it copy and save some packet randomly with putting the flag to inform CN to return these packets back. Therefore MN is able to compare these two packets (saved before and came back from CN),

whether are same or not. If packets are not same based on the data, MN can decide to use encryption to protect the data. If attacker change the flag that means the MN will not receive the selected packet form CN or will receive unselected packet, the MN will start encryption. The encryption key can be sent to CN or CNs during RR procedure. Fig. 1. shows the signal flow of the proposed method.

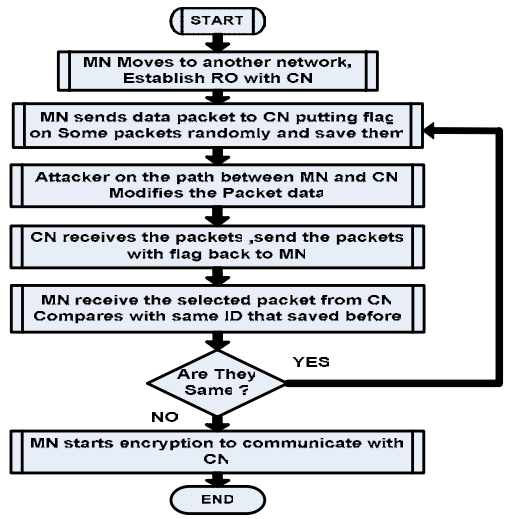


Fig. 1. Signal Flow of Proposed Method

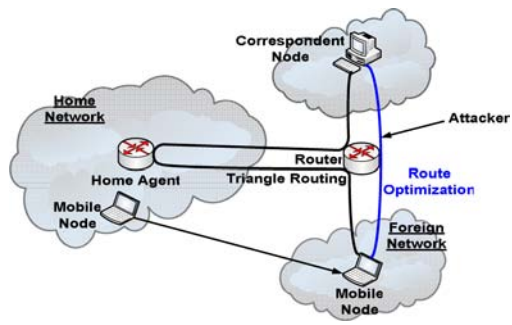


Fig. 2. Test-bed Design and Architecture

Due to complication and problem in using IPsec and encryption methods in RO, encryption is used only if an attacker is found which is suitable for delay sensitive applications. Buffering the packets by MN is randomly, it is concluded fast transmission and because of the necessarily of CN to return back the selected packet, is not leading to increase the network traffic on the large networks.

We setup a Mobile IPv6 Test-bed with IPsec and Route Optimization enabled to examine the performance of our security method. IPsec is enabled on HA and MN to avoid forged messages. The Test-bed is composed of hardware, software and network

analysis tools to capture and monitor the packet flow and content of data. The attacker is programmed using middleware and applies on the Test-bed to show how it can affect on the packet and how this method can prevent it. This attacker is able to modify the packet from and to the CN.

The implemented network Test-bed consists of four computers. Two of them assume the roles of the CN and MN, respectively, one Home Agent and one Router are configured as IPv6 capable router. PC-based software router implementation is used instead of commercial IPv6 router in order to have more flexibility and possible to run middleware program. The design and architecture of the proposed scheme are shown in Fig. 2.

During this research, because of compatibility and MIPv6 functionality, Fedora core 5 (New kernel 2.6.16) was used as operating systems with MIPL (Mobile IPv6 for Linux) version 2.0.2 that is the most recent release for MIPv6 implementation and fully RFC 3775 compliant. HA is PC with one Ethernet network card and one wireless card, Router has two Ethernet network card and one wireless card, and CN has one Ethernet network card. MN is Laptop with wireless LAN.

6 Result and Discussion

From the results of the Test-bed it is shown that how attacker can modify the packets and this method prevents it, as well as performance of the security proposed method and packet flow. Prior to the Test-bed performance measurement, Network Time Protocol (NTP) is used to synchronize the time on MN (packet generator) and CN (packet receiver). MN generates 100-byte-long unicast packets periodically in every 100ms. To prevent the attackers from capturing and modifying the data, we used an encryption method to do so. There are strong methods of authentication involving public key cryptography that can be used. Our proposed scheme involves the using of Transport Layer Security protocol with both peers knowing a secret key.

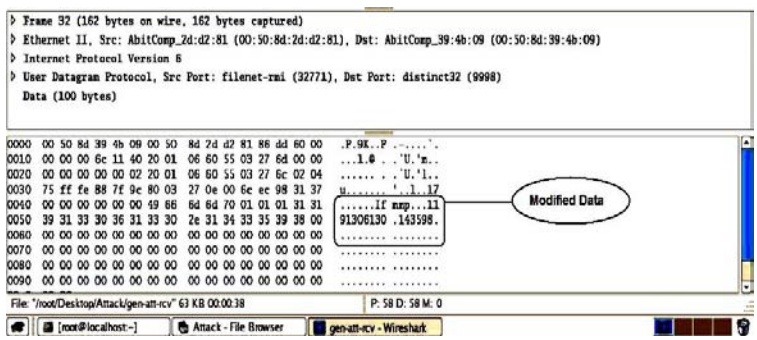


Fig. 3. Modified Packet Received by CN

Fig. 3. shows the modified data packet that MN has sent the “Hello” packet and attacker modified it before receiving by CN. This packet is captured by Wireshark software on router.

It can be seen from Fig. 4. that until 17s, the delays are same for both conventional and enhanced method. After 17s the MN detects the attack until 20s. The MN at 20s starts data encryption to prevent attackers from modifying the packets. The delay in proposed method is because of data encryption.

Fig. 4. Packet Flow on Conventional and Proposed Method

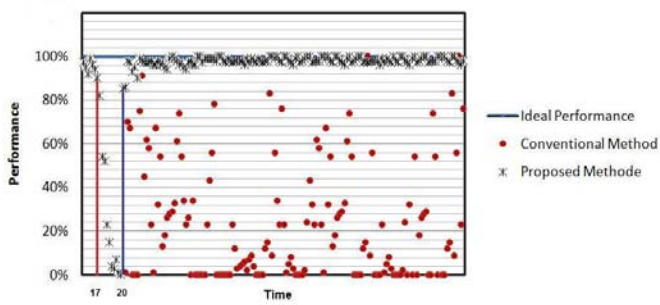


Fig. 5. Performance Comparison of the Conventional and Proposed Method

It can be seen from Fig. 5., before starting attack at 17s it seems the packets flow and packets lost are normal and the performance is near to ideal performance on the wireless networks. It can be seen that the performance is decreased on the conventional method when attacker start the attack. For enhanced method, the performance is rapidly increased after MN starting encryption the packets at 20s.

7 Conclusion

This paper gives an overview of Mobile IPv6 security, proposes a new security method algorithm, development of the Test-bed for evaluating the performance and effectiveness of the proposed method in comparison with the conventional method. The proposed method gives advantages of safe communication in terms of data security in Route Optimization Mobile IPv6 networks.

References

1. Johson, D., Perkins, C.: Mobility Support in IPv6, RFC 3775 (2004)
2. Arkko, J.: Using IPsec to Protect Mobile IPv6 Signaling between Mobile Nodes and Home Agents, IETF RFC 3776 (2004)
3. Ren, K., Lou, W., Zeng, K., Bao, F., Zhou, J., Deng, R.H.: Routing Optimization Security in Mobile IPv6. Computer Networks 50(13), 2401–2419 (2006)
4. Elgoarany, K., Eltoweissy, M.: Security in Mobile IPv6: A Survey, Information Security Technical Report, vol. 12(1), pp. 32–43 (2007)
5. Perkins, C.: Securing Mobile IPv6 route optimization using a static shared key, RFC 4449, IETF (June 2006)
6. Aura, T.: Mobile IPv6 security. In: Christianson, B., Crispo, B., Malcolm, J.A., Roe, M. (eds.) Security Protocols 2002. LNCS, vol. 2845, pp. 215–234. Springer, Heidelberg (2004)
7. Kent, S., Atkinson, R.I.: Encapsulating security payload (ESP), RFC 4303, IETF (December 2005)
8. Kent, S.: IP authentication header, RFC 4302, IETF (December 2005)
9. Tuomas, A.: Designing the Mobile IPv6 security protocol. Annals of Telecommunications (Special issue on network and information systems security) (March–April 2006)
10. Nikander, P., Aura, T., Arkko, J., Montenegro, G.: Mobile IP version 6 Route Optimization Security Design Background, Expired IETF Internet Draft (2003)
11. Nikander, P., Aura, T., Arkko, J., Montenegro, G.: Mobile IP version 6 (MIPv6) Route optimization security design. In: Proceedings of the IEEE Vehicular Technology Conference Fall 2003, Orlando (2003)
12. Aura, T.: Cryptographically Generated Address (CGA), IETF RFC 3972 (2005)
13. Harkins, D., Carrel, D.: The Internet key exchange, RFC 2409, IETF (November 1998)

Adaptive End-to-End QoS for Multimedia over Heterogeneous Wireless Networks

Ouldooz Baghban Karimi, Mahmood Fathy, and Saleh Yousefi

Iran University of Science and Technology, Computer Engineering Department
Tehran, Iran
ouldoozbk@comp.iust.ac.ir, {mahfathy,syousefi}@iust.ac.ir

Abstract. Colossal numbers of ways have been proposed to decrease delay, jitter and loss in wireless networks for good user perceived quality in video over internet. This paper studies the multimedia over heterogeneous wireless networks, requirements and problems and proposes a new scheme to overcome the obstacles. The proposed scheme, takes into account the effects of Application Level Wireless Multilevel ECN marking, thus helps us overcome the congestion/loss mistake problems. For handoff, handover and lossy link problems, it is considered that a freezing mechanism is in use in application layer and assumed that the upper layers can be aware of disconnection periods to make the rate adaptation decisions. Also a new scheme has been added to receiver to gracefully degrade the quality when no other action is available to combat the long delays without data which is caused by handoffs and wireless temporary disconnections.

Keywords: Video over wireless, heterogeneous, graceful degrade, quality adaptation, rate adaptation.

1 Introduction

There are three main impediments in QoS support: Variable network performance, network congestion, and unpredictability of the bandwidth availability in the network.

For an end-to-end QoS support, there always has been a choice of TCP or UDP. UDP is connection-less and does not maintain packet numbering, timing and network state information. There have been a number of ways proposing UDP in companion with RTP for time stamping and sequence number and using RTCP for obtaining network information to support QoS over wireless networks which is the preferred method for multimedia transmission over wireless networks [1]. The main advantage of this way is that UDP does not maintain packet loss and retransmission and it does not include any kind of congestion control mechanisms which take a lot of time, especially with loss/congestion mistake scenarios in wireless networks. But as the compression volume increase, the need to retain lost packets increase. On the other hand TCP works effectively over traditional networks but it does not work effectively for multimedia services as its performance degrades due to window size decrease in the response to packet loss resulting from the congestion.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 160–167, 2008.

TCP shows even worse performance over the wireless networks due to five main problems: Limited bandwidth, long round-trip times, random losses, handoffs, and short flows services.

There are number of ways to solve these problems including pure end-to-end protocols, link layer protocols, split connection protocols, soft-state transport-layer caching protocols, and cross layer signaling protocols [1, 2] but none of the proposed methods could solve all the TCP regarding problems in wireless networks.

In this paper we propose a way to overcome the video streaming problems in mobile wireless networks and provide an algorithm that shows better QoS and multimedia delivery over wireless networks. The proposed mechanism aims at adaptive end-to-end QoS support so it is classified in pure end-to-end protocols category.

2 QoS Support Obstacles

We assumed the main problems of QoS support over wireless networks are random losses and handoffs as well as congestion. There are several proposed methods to overcome these problems. The main problems and shortcomings of proposed features available till now are discussed here in detail.

Congestion. In the traditional TCP, packet loss is the main indicator of congestion events. In such networks, where the link error is very low, this mechanism would show acceptable results but when used in wireless networks, it may cause starting the congestion control mechanisms in the case of random packet losses. One of the famous ways to solve this problem is explicit congestion notification. ECN [2, 3, 4] marking allows one to explicitly notify the receiver and therefore in ACK packets the sender, about the congestion and how to act accordingly. But for supporting QoS over the wireless networks it is not enough to have one bit indicator of congestion.

Handoff and Temporary Disconnections. In the mobile networks there is a disconnection gap while a mobile host moves from one cell to another. In this disconnection period, all of the packets in transition will be lost. In the disconnection period the packets or their ACK packets are going to be lost and sender may want to resent them that will take some of the sender's time; or there may be the possibility that sender decrease its congestion window size in transport layer that would affect video quality tremendously. The problem gets worse when considering heterogeneous wireless environment. In such an environment, it is considered that a handoff would occur not only between two adjacent cells of the same service, but also among different service areas which would cause longer handoffs. This means that heterogeneous wireless networks need handoff and disconnection aware mechanisms for QoS support.

Multimedia over Wireless. Multimedia and more specifically video is very sensitive to network QoS parameters. User perceived quality is one of the meters used as an indicator of quality of transmitted video over the wireless networks. It is measured by availability of frames to playback at the receiver. In the case of long temporary disconnections caused by handoffs and handovers, the playback buffer used to maintain delay and jitter, runs out of packets to show and this will affect the user perceived quality dramatically.

3 AWEQ

Our proposed algorithm (Adaptive Wireless End-to-End QoS - AWEQ) tries to solve the problems discussed above with WMECN in network layer, UDP mechanism in transport layer, disconnection signaling mechanism in link layer, and graceful degrade, rate adaptation and quality adaptation in the application layer.

Multilevel ECN-like mechanism. For getting a precise feedback from the network, Application-level Wireless Multi-level ECN marking is used [3]. The used process is much like discussed in [5, 6]. As it is proposed in [3], AWMECN uses two bits that are being specified for the use of ECN in the IP header to indicate four different levels of congestion instead of binary feedback provided by ECN. Four levels obtained by two bits are assumed for following feedbacks: "00" for non-ECN capable feedback, "01" for no congestion notification, "10" for mild congestion notification and "11" for severe congestion notification.

Three different thresholds have been proposed on the red queue, minimum threshold, maximum threshold and middle threshold. When the queue size is lower than minimum threshold no packet is marked. When the queue size is between minimum and middle threshold, packets are marked with "10" with maximum probability $P_{\max}$. The rest of packets remain unmarked. When the queue size is between middle and maximum thresholds, packets are marked with "11" with maximum probability $P_{\max}$ and the rest of packets are marked with "10". After the maximum threshold, all the packets are marked with "11". All these values are set in an application level header of packet to be analyzed by the application as proposed in [3].

Handoff and Freeze Mechanism. We assume that the UDP is used for transport layer mechanism. Sequence numbering and time stamping is done in application level. We assume a freezing mechanism run at application level. The first step to this was to use a freeze-TCP like mechanism, namely ATCP, run at the transport layer [7]. Then we modified the algorithm and changed the decision making center to be in application layer. We just assumed that the link layer would signal and aware the adaptation algorithm in application layer from the probable disconnection event and the application layer would adapt its sending rate and quality accordingly. Note that the possibility of future disconnections which could be estimated by signal strength in mobile host is used instead of disconnection event signal. The transport layer mechanism would be a simple UDP. The freeze mechanism in application layer would result in bigger throughput due to the fact that it will never encounter the slow start, retransmission and congestion window problems.

The proposed mechanism would just decrease the sending rate at the sender to zero in the probable handoff and disconnection events. Then it will restart sending the video with the previous rate when the disconnection duration is over. All the decisions are made in application level in the receiver so the sender would need to be aware of what to do. Since the disconnection probability is announced in advance to the occurrence of event, receiver could inform the sender about the disconnection in the last talk. We assume that both the sender and the receiver could be mobile and wireless hosts and assumed that in the event of disconnection origination from sender side, sender would also inform the receiver from the handoff and disconnection

possibility and would decide by itself to decrease the sending rate to zero. After the disconnection is resolved, the sender would check new state with receiver and act accordingly.

Rate and Bandwidth Evaluation. Available bandwidth is the minimum unused bandwidth of any of the links along the path between sender and receiver. If the transmission rate of the flow is lower than the available bandwidth or equal to it, then the arrival rate is equal to transmission rate. Otherwise, the arrival rate will be lower than transmission rate. There is only one value of the transmission rate that the two are equal. For other cases, the link capacity must be known to assess the available bandwidth.

The arrival rate at the destination is a function of transmission rate, the capacity and the available bandwidth of all links along the path [5]. No single bottleneck could determine the arrival rate. For end-to-end measurement of available bandwidth a series of periodic flows is transmitted between the end points. For each flow the receiver analyzes one-way delay variations of the packets to determine whether the transmission rate is higher or lower than the available bandwidth. A set of video packets can be treated as a packet train and used to obtain an estimate of asymptotic dispersion rate, which are referred to as arrival rate at the client. The measurement is done over a moving window of certain size. Thus, the number of packets may vary from one estimate to another. This estimate is then used to draw a conclusion about the available bandwidth. We assumed the rate of the packets received is calculated using (1) which add together all the received packets p_i during the Δt_r time interval.

$$r_r = \frac{1}{\Delta t_r} \sum_i p_i \tag{1}$$

The calculation is then modified and enhanced with the $0 \le \alpha \le 1$ factor which is used for taking into account the previous values of received packers in the present calculation.

$$\hat{r}_r = (1-\alpha) r_r + \alpha r_r \tag{2}$$

The video arrival rate is compared against the transmission rate and conclusion is drawn. The conclusion is in two state domains which show one of followings: available bandwidth is lower than needed by the video stream or is equal or higher then needed by video stream.

Rate Adaptation. After the available bandwidth and transmission and arrival rates have been calculated and ECN is taken to know about the network state and handoff and disconnection events are tested. After all, the playback buffer occupancy for the next Round Trip Time is estimated using the sum of current value of buffer size plus the received packets in next RTT minus the used packets in next RTT:

$$buf\,(t + RTT) = buf\,(t) + \sum_{i=t+1}^{t+RTT} \hat{r}_r(i) - \sum_{i=t+1}^{t+RTT} u_i \tag{3}$$

For a given level of quality, a default schedule using a fixed transmission rate and minimizing the client buffer requirement is prepared ahead of time based on knowledge of video content. The receiver can toggle its need between r_h, a rate higher than

the default rate and r_l, a rate less than the default rate without need to change quality level. One can simply toggle or smoothly go the way between these two values.

The time Δt_h in which the sender could increase the sending rate is related to the size of network backlog in time t in which the request is made, and the playback buffer occupancy:

$$\Delta t_h = \max\left(t \le i \le N-1 : buf(t) + (i-1) r_h - \sum_{j=t}^{i} u_j \le B \right) - t \tag{4}$$

The lower rate is requested when the playback buffer occupancy is reached a predefined value. In this time the lower rate is used to compensate the higher rate before the playback buffer would overflow. In contrast to the mechanism used in [5], in our mechanism there is no need to use this value after all high rate send periods.

$$(r_h - r)\Delta t_h = (r - r_l)\Delta t_l \tag{5}$$

Quality Adaptation. There are four different known ways for quality adaptation: adjusting the compression ratio of an on-line encoder, switching among different pre-encoded versions, transcoding a pre-encoded version and dropping a layer of a hierarchical encoding, scheme. In this paper, quality adaptation is assumed to be done in the sender side of the connection by switching between different pre-encoded versions of video requested.

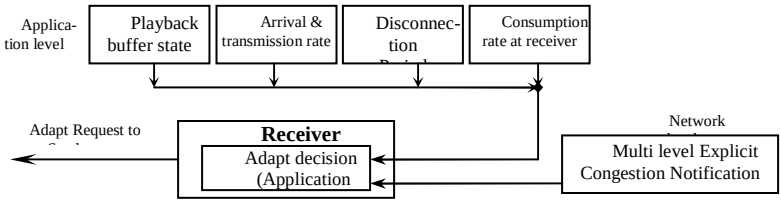


Fig. 1. Adaptation mechanism

Graceful Degradation. Rate and quality adaptation are done in a way that we could have the enough frames in the playback buffer to show to the user, but when the video size increases and the time interval of the connection becomes long, in the cases that the playback buffer underflow, the problem of running out of frames to show becomes serious. In this case, when the buffer underflow and there seems to be a congestion or a wireless oriented disconnection in the network, the algorithm tries to impose a maximum playback buffer use rate on the user. This can be done with a delay between using two subsequent frames which is calculated in (6).

$$\textit{Playback Delay (i)} = \min(10ms, [TH - (Np(i) + Ra/Ru)] \times C) \tag{6}$$

TH is the minimum threshold used for playback buffer. *Np* is the number of packets in playback buffer while using the *i*th packet. *Ru* is the usage rate of playback buffer frames. *Ra* represents the arrival rate of packets at the user and C is the playback delay constant which is considered to be 1ms for our scheme. The delay should be found such that affect the video quality in an extremely controlled way. For this

Table 1. Decision making based on MECN value, handoff possibility and receiver buffer state

Ar < Tr	ECN	Handoff	Buff (t+RTT)	Action	
				Receiver	Sender
-	-	Yes	Overflow	None	Rate =0
-	-	Yes	Underflow	Graceful Degrade	Rate =0t & Dec Quality
-	-	Yes	OK	Graceful Degrade	Rate = 0
False	00	No	Overflow	None	Dec Rate (r→ rl)
False	00	No	Ok	None	Inc Rate (r→ rh or r→previous r)
False	00	No	Underflow	None	Inc Rate (r→ rh or r→previous r)
False	00	No	OK	None	Inc Quality or
True	00	No	OK	None	None
True	00	No	Underflow	Graceful Degrade	None
True	10	No	OK	None	None
True	10	No	Underflow	Graceful Degrade	Dec Quality
True	11	No	-	Graceful Degrade	Dec Quality
True	-	No	Overflow	None	Dec Rate (r→ rl)

purpose an upper bound for this delay is assumed to be 10ms, slightly more than 5ms which is acceptable in 12fps playback.

4 Simulation and Results

We implemented the proposed algorithm using NS-2 network simulation environment. The simulation was tried to meet the actual delay, bandwidth and rates used in mobile communications. The added parts to the NS-2 network simulation environment to implement this characteristics and the AWEQ method are shown in figure 2. We considered a two-part scenario for the test. The first part of the test scenario was to evaluate the AWMECN performance.

Avoidable retransmitted data and connection duration are used to be the main metrics used for evaluation of the mechanism (Fig.3). Transmission duration is also used for both evaluations of performance and average power consumption estimations. Different scenarios used for evaluation of the mechanism, show an average reduce of 1.9% in transmission duration compared with similar methods. The proposed method shows a good communication duration and low extra data so leading to an efficient power consumption scheme.

The second part of the test was the evaluation of the whole mechanism. We considered the video traffic is also CBR traffic with the rate of 0.1Mbps, as discussed in [5]. Playback buffer occupancy and variations are used as indicator for acquiring the user-perceived quality.

The overall performance of the proposed algorithm was measured by means of the playback buffer as an indicator of user-perceived quality. As it is obvious in figure 4, the variations in playback buffer are very low in the proposed mechanism and it has not underflow below a definite threshold while the non-adaptive algorithm used in comparison, which is assumed to be the normal non-adaptive application over the same underlying layers, let the playback buffer totally underflow to the value of zero. Frame size is considered to be 500 bytes for simplicity.

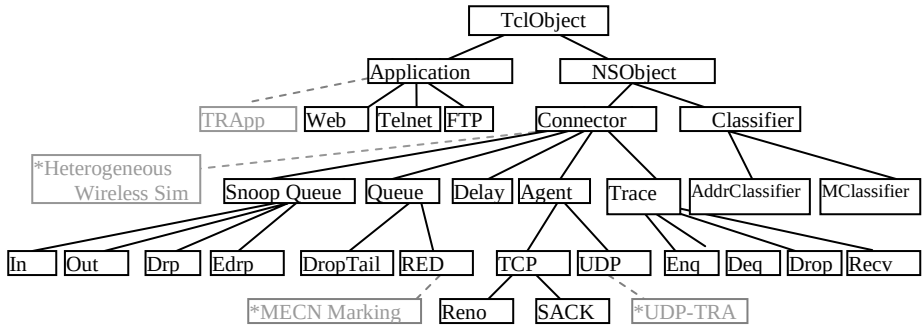


Fig. 2. Parts added to NS-2 for simulation of the AWEQ algorithm

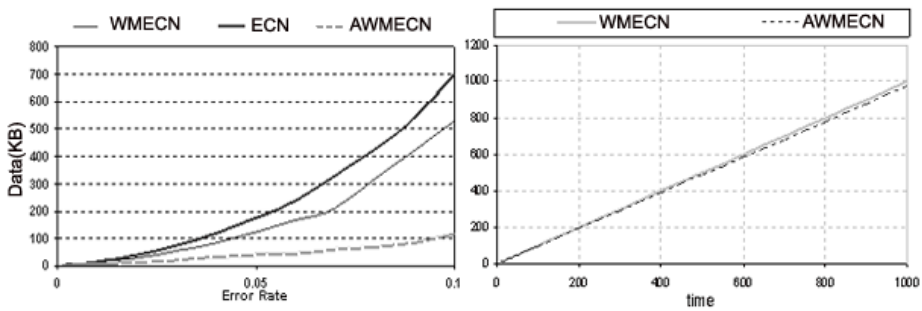


Fig. 3. (a) Avoidable retransmitted Data **(b)**Transmission duration

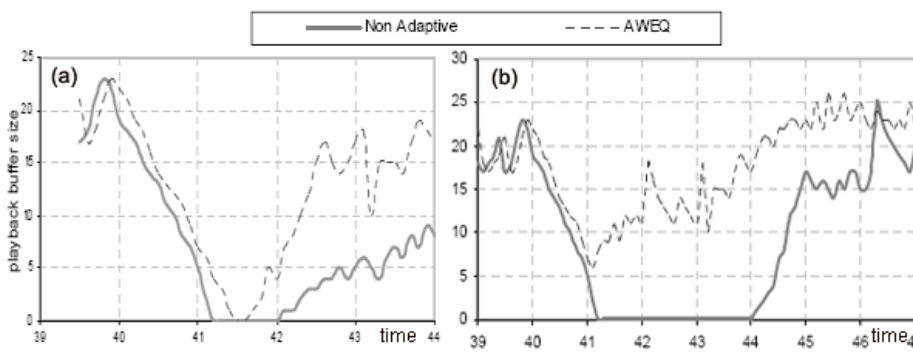


Fig. 4. Playback buffer occupancy during (a) disconnection (b) congestion

5 Conclusion

In this paper we studied a way to overcome the video streaming problems in mobile wireless networks and provided an algorithm that shows better QoS and multimedia delivery over wireless networks.

We conclude that it is a better engineering decision to choose best protocols in each layer and using adaptation in application layer. We see AWEQ as a promising definition of such an algorithm and hope to see it incorporated in the standard implementations for video delivery over heterogeneous wireless networks.

References

1. He, X., Xu, L., Liu, M.: TCP Performance Evaluation over Wireless Networks. In: Canadian Conference on Electrical and Computer Engineering, vol. 2, pp. 983–986 (2004)
2. Rakocevic, V.: Congestion Control for multimedia applications in the wireless Internet. International Journal of Communication Systems 17(7), 723–734 (2004)
3. Baghbankarimi, O., Fathy, M., Yousefi, S.: Application level Wireless Multi-level ECN for Video and Real-time Data. In: ICN 2006, IEEE proceedings, p. 137. IEEE Press, Los Alamitos (2006)
4. Goff, T., Moronski, J., Phatak, D.S.: Freeze-TCP: A true end-to-end enhancement mechanism for mobile environments. In: IEEE Infocom, pp. 1537–1545. IEEE, Los Alamitos (2000)
5. Kusmierek, E., Du, D.H.C.: Streaming video delivery over Internet with adaptive end-to-end QoS. The Journal of Systems and Software 75(3), 237–252 (2004)
6. Kunniyur, S., Srikant, R.: End-to-End Congestion Control Schemes: Utility Functions, Random Losses and ECN Marks. IEEE Transactions on Networking 11(5), 689–702 (2003)
7. Singh, A.K., Iyer, S.: ATCP: Improving TCP performance over mobile wireless environments. In: IEEE 4th international workshop on mobile and communication networks, pp. 239–243. IEEE Press, Los Alamitos (2002)
8. Reddy, V., Sharama, V., Suma, M.B.: Providing QoS to TCP and Real Time Connections in the Internet. Queuing Systems 46, 461–480 (2004)
9. Ding, W., Jamalipour, A.: A New explicit loss notification with acknowledgement for wireless TCP. In: Proceedings of IEEE PIMRC 2001, San Diego, CA. IEEE Press, Los Alamitos (2001)

A Blocking Reduction Scheme for Multiple Slot Cell Scheduling in Multicast Switching Systems

Fong-Cheng Lee, Wen-Fong Wang, and Jung-Bin Shih

National Yunlin University of Science and Technology
{G9213716,WWF,G9417716}@yuntech.edu.tw

Abstract. In this paper, we propose a multicast switch called BRMSCS switch consisted of shared memory, crossbar fabric and the scheduler BRMSCS (Blocking Reduction Multiple Slot Cell Scheduler). Our goals are to reduce the blocking situation in the scheduler and to guarantee free of memory access conflict (MAC), that is, no more than two output ports would access the different cells which come from the same input port. We separate the BRMSCS switch into two parts: the data section and the control section. The header of each incoming cell is sent to the control section and scheduled by BRMSCS. To meeting our goals, the RRMSCS scheduler quickly inserts the address cell into scheduling table and fills the scheduling table as full as possible. The simulation results show that the BRMSCS scheduler can efficiently insert the address cell into the MAC free location of scheduling table and has the feature of reducing blocking.

Keywords: Multicast Switch, Multicast Scheduler.

1 Introduction

Switches can basically classify into input queueing (IQ) switches and output queueing (OQ) switches. The ideal OQ switches can achieve the highest throughput and the packet latency is similar to the M/D/1 queueing system. However, the OQ switches are impractical, when the input line rate or port number increasing. For an input queuing (IQ) switch, Karol et al. had shown that the delay-throughput is limited to 58.6% [1]. This is caused by HOL (head of line) blocking. Adopting VOQs (virtual output queues) can completely eliminate the HOL blocking [2]. For multicast traffic, traditional unicast switch replicates a multicast packet to multiple unicast packets before entering the switch system. This would cause seriously input and output blocking problem. As the result, switching multicast traffic by a traditional unicast switch causes two problems. First, the size of memory requirement is increasing. Second, the utilization of the bandwidth is decreasing. For the above reasons, recently high performance and low cost multicast switch architecture and scheduling algorithm were proposed.

In [3], ESLIP scheduler adopting VOQ and additional multicast queues is proposed. Chao et al. proposed a serial DRRM scheduler and token tunneling method to reduce the information changing inside the switch [4][5]. In [6], TATRA scheduler was proposed. TATRA scheduler is easy to implement however more than two output

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 168–175, 2008.

ports would access the different cells which come from the same input port. We call this situation memory access conflict (MAC). Chen et al. proposed that combining copy network, non-blocking routing network, and MSCS scheduler to switch multicast traffic [7]. The MSCS scheduler will select as many as possible cells to the copy network and the copied cell will through the non-blocking routing to the output. However, there is blocking happening in the MSCS scheduler. In this paper, we propose a scheduler called BRMSCS (Blocking Reduction Multiple Slot Cell Scheduler). Our goals are to reduce the blocking situation in the scheduler and to guarantee memory access conflict (MAC) free. By using larger recoding tables, the blocking can efficiency reduced. The simulations results also show that BRMSCS scheduler has the features of reducing blocking and efficient scheduling multicast traffic. The rest of this paper is organized as follows. In Chapter 2, we introduce the model of proposed BRMSCS switch. In Chapter 3, we make a detail description and illustration of the BRMSCS scheduler. The simulation results of the BRMSCS scheduler are shown in Chapter 4. The conclusions are made in Chapter 5.

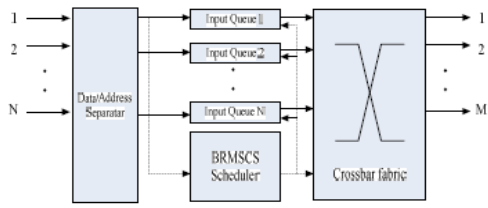


Fig. 1. The NxM BRMSCS switch system

2 Switch Model

The BRMSCS switch can be describe as follows: buffering strategy, switching strategy and scheduling strategy. As shown in Fig.1, BRMSCS switch is organized by input queues, crossbar fabric and BRMSCS scheduler. It separates the control and data section in its internal architecture. The data section is only responsible to store the cell's payload. The control section is only responsible to schedule the output order for incoming cell herders. Before entering the crossbar fabric, the cell header and corresponding cell's payload will be rebounded. In following, we make a detail description of BRMSCS switch architecture.

2.1 Buffering Strategy

N*N VOQs (virtual output queues) in input ports to eliminate the HOL blocking problem. However, for multi-cast switches, there must use N*(N_2-1) VOQs to completely eliminate the HOL blocking problem. This will make the multi-cast switches impractical. For saving cost and efficiently using the memory, the BRMSCS switch adopts share memories as input queues, as shown in Fig. 1. When a cell arrives, the Data/Address Separator will assign an available address for storing the cell's payload in the input share memory. At the same time, the cell's header will be copied as an

address cell and sent to the BRMSCS Scheduler. An address cell involves the following information: the source port, the destination ports, the payload address, and the arrival time. The payload address is used of indicating the physical address of shared memory of cell's payload. The source port, destination ports are used of scheduling and the arrival time is only used of scheduling multicast traffic. There is an additional table to record the number of fanout of arriving cells. When a cell departures from the switch, the corresponding fanout record will decrease by one. When the fanout record becomes to zero, the corresponding cell's payload will be removed from the share memory.

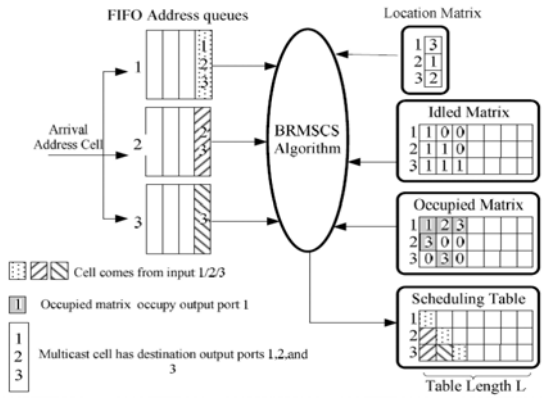


Fig. 2. The BRMSCS Scheduler

2.2 Switching Strategy

The BRMSCS switch adopts a crossbar fabric. The major reasons are it is non-blocking and has native property for multicasting. By controlling the N*N cross-points of the crossbar fabric, it can provide N non-blocking paths from inputs to outputs. There are two constrains while using crossbar fabric: a) each input port can only sent at most one cell into the fabric and b) each output port can receive at most one cell from the fabric. For maximum the switch performance, we need additional scheduler for scheduling the incoming cells to outputs and also not violate crossbar constrains. For multicast switching, a multicast cell can be sent to different output port at the same slot without against the crossbar constrains.

2.3 Scheduling Strategy

The BRMSCS scheduler is organized as Fig. 2. When an address cell arrives, it will be queued in the corresponding FIFO address queues. Only the HOL address cells in each address queue can be scheduled by the BRMSCS algorithm. There is blocking happening in the HOL of FIFO Address queues. By using larger tables and Procedure Insert of BRMSCS algorithm can efficiency reduce the blocking. For following the crossbar constrains, the scheduling results must guarantee the MAC is not happening. The BRMSCS algorithm guarantees that it finds the MAC free location in the

scheduling table. The scheduling table is used to store the scheduling results. At the end of each run of scheduling, each address cell at the first column of each scheduling table will be removed. After rebounding to its referring payload, the cell will be transferred from the input queue through the crossbar fabric to the output port(s).

3 The BRMSCS Algorithm

3.1 Definitions

1. $C_{i,j}$: an incoming cell comes from source port i and destines to output port j. If it is a multicast cell, it has multiple output ports.
2. $ID(C_{i,j})$: the id of a address cell $C_{i,j}$.
3. ST(x,y),Scheduling Table: where x and y represent the row x and column y of the scheduling table. $ST(x,y)=ID(C_{i,j})$ means the column y and row x of scheduling table is occupied by the address cell $C_{i,j}$ which id is $ID(C_{i,j})$.
4. LM(n), Location vector: LM(n) is the last location of the address cell in the scheduling table which comes from input port n.
5. OM(x,y), Occupied matrix: where x and y represent the row x and column y of the occupied matrix. As OM(x,y) is not null and OM(x,y)=k, there is at least one location of column y of scheduling table is occupied by address cell which comes from input x and destined to output k.
6. T(ST(x,y)): the arrival time of the address cell $C_{i,j}$ which id equals to ST(x,y).
7. IM(x,y), Idled matrix: where x and y represent the row x and column y of the idled matrix. IM(x,y)=True means the row x and column y of scheduling table is occupied, otherwise IM(x,y)=False.

[The BRMSCS Algorithm]

```
/* Procedure Insert: */
For i=1 to N
   /*Referring to the LM and finding a MAC free location */
   If HOL cell of Address queue(i) ≠ NULL
      Let Cmn = HOL cell of Address queue(i)
      For each fanout of Cmn
         For k= LM(i)+1 to L
            If IM(n,k) is False
               Set ST(n,k) = ID(Cmn)
               Set IM(n,k) = True
               Set OM(m,k) = n
            EndIf
         EndFor
      EndFor
   EndIf
End For

/* Procedure Search and Exchange */
For i=1 to N
  For j=1 to L
      /*If Scheduling table is null, finding the eligible
cell*/
      If IM(i,j) is False
         For k=j+1 to L
```

```
            If IM(i,k) is True
               Set Cm,y is the corresponding Cell of IM(i,k)
               Check eligiable:
               (1) OM(Cm,y) = null)
               (2) (OM(Cm,y)≠ null) AND (T(ST(i,k)) =
T(ST(OM(n,k),y))
               If eligible
                  Swap (S(I,k),ST(I,j))
                  Update OM and IM
            EndIf
         EndFor
      EndIf
   EndFor
EndFor
```

3.2 The BRMSCS Algorithm

The BRMSCS algorithm involves two procedures: 1) Insert and 2) Search & Exchange. The procedure Insert quickly finds a MAC free location of scheduling table and inserts HOL address cell to the scheduling table. The procedure Search and Exchange is used to find an unoccupied location and exchange the first eligible address cell to the unoccupied location. An eligible address cell is the address cell behind the empty location and guarantees MAC free after exchanging the eligible address cell to the empty location. The following description will explain how these two procedures work.

Procedure 1) Insert:

In Insert procedure, one of the FIFO address queue is selected by the local round-robin pointer, and the HOL address cell Cij of the selected queue will be scheduled

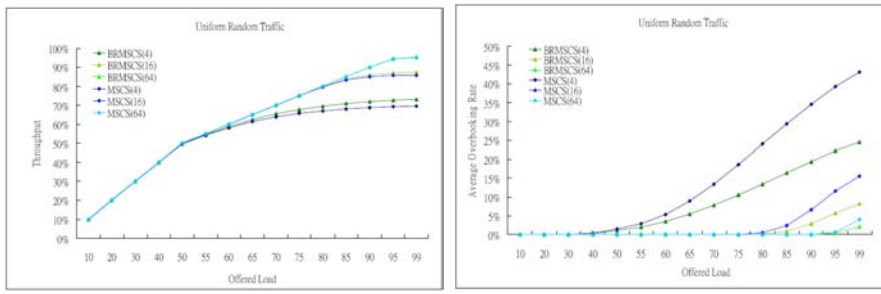
by the Insert procedure. The Insert procedure will first refer to the LM(i) as the starting location i.e. LM(i)=p and search the first unoccupied location from ST(j,p+1). This procedure will be executed at most L-p times, where L is the scheduling table length. For keeping fairness, at the end of Insert procedure, the round-robin pointer increases one. Note that the LM is not updated in this procedure; it will be updated at the end of procedure Search & Exchange.

Procedure 2) Search & Exchange:

To maximum the switch performance, the scheduling table must be filled as fully as possible. The Search & Exchange procedure is try to fill the unoccupied location of scheduling table by exchanging the address cell from occupied location to former location without MAC. The Search & Exchange procedure will search the first unoccupied location from each column. If the unoccupied location ST(x,y) is founded, searching the eligible address cells Cmx from the column y+1 in row x. To guarantee MAC free, the eligible address cells Cmn located in ST(n,p) must satisfy following two conditions.

Condition 1: (OM(Cm,y) = null)
Condition 2: [(OM(Cm,y)≠ null) AND (T(ST(n,p)) = T(ST(OM(n,p),y))]

If the address cell, located after column y+1 in row x, is unicast traffic, the procedure Search & Exchange only verifies the condition 1. OM(Cm,y) = null means that



(a) (b)

Fig. 3. Throughput (a) and Average Overbooking Rate (b) under Uniform Radom Traffic

the input port m of Cmn is not appeared in the column i of scheduling table, thus Cmn is eligible to move to unoccupied location ST(n,i). If the verified address cell is multicast traffic, the procedure Search & Exchange both verifies the condition 1 and condition 2. For example, a cell D already occupied ST(x,z), $z \neq y$, however, the arrival time of the cell C and cell D are same. This means eligible cell C and cell D are the same fanout of a multicast cell. Moving eligible cell C to the unoccupied location still guarantee MAC free.

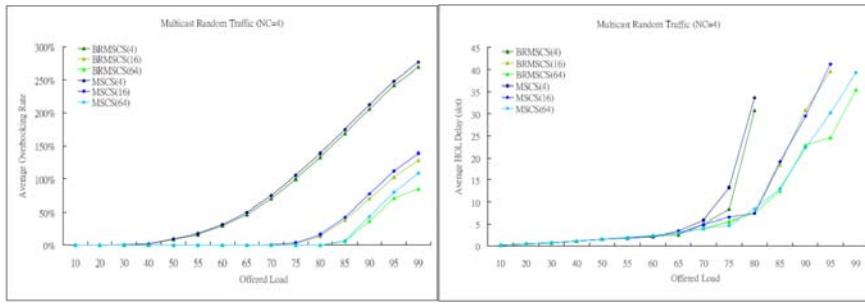
4 Simulation and Results

In this chapter, we will present the simulation results of BRMSCS multicast switch. The search depth L is set 4, 16, and 64. The length of scheduling table of MSCS and BRMSCS is set L and 2L. For example, BRMSCS(4) means the search depth is 4 and length of all recording table of BRMSCS is 8. The switch size is set to 32x32 and the simulation time is set to 1000000 time slot. We also assume there is no cell lost in the switch that is the buffers are infinite. Three kinds of traffic models in [9] are used to evaluate the performance of proposed BRMSCS switch: Unicast Random Traffic, Multicast Random Traffic, and Multicast Bursty Traffic. The average fanout (NC) is setting 4 and 8 with exponential distributed. The simulation diagrams include as following:

- Throughput
- Average HOL delay: the time starting from address cell becoming the HOL address cell until its last fanout leaves from the scheduling table.
- Average overbooking rate: When the fanout of HOL address cell can not be scheduled into the scheduling table in one time slot, the overbooking is occurring. The overbooking rate is the ratio of number of successful scheduled fanout to unsuccessful scheduled fanout. It can be used to evaluate the blocking frequency of BRMSCS and MSCS scheduler.

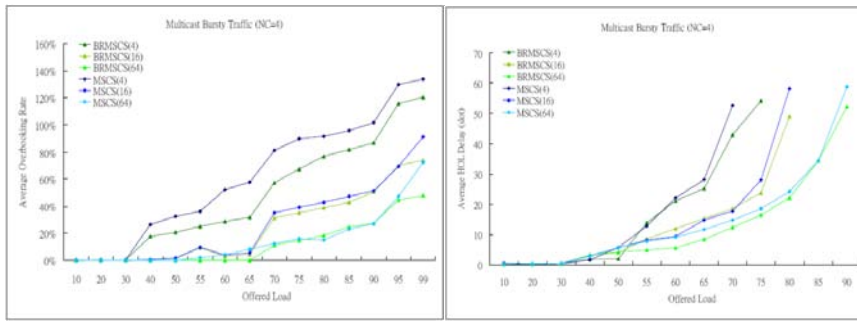
4.1 Uniform Random Traffic

Fig.3 illustrates the simulation result of BRMSCS and MSCS scheduler under uniform random traffic. We can obviously see that, in Fig.3(b) and Fig.3(a), the average



(a) (b)

Fig. 4. Average Overbooking Rate (a) and Average HOL Delay (b) under multicast random traffic (NC=4)



(a) (b)

Fig. 5. Average Overbooking Rate (a) and Average HOL Delay (b) under multicast bursty traffic (NC=4)

HOL delay and throughput of BRMSCS is better than MSCS. The most important reason is BRMSCS reducing the blocking which is happening in the FIFO address queue.

4.2 Multicast Random Traffic

Fig.4 illustrates the simulation results of BRMSCS and MSCS scheduler under multicast random traffic with NC=4. Obviously, the larger search depth, the better delay performance and lower overbooking rate. Also the BRMSCS performs better delay and overbooking rate than MSCS. As the average fanout, NC, is increasing, the both scheduler switch must run deeper search for high performance.

4.3 Multicast Bursty Traffic

Fig.5 illustrates the simulation result of BRMSCS and MSCS scheduler under multicast bursty traffic with NC=4. Obviously, both BRMSCS and MSCS perform poor. Even increasing the search depth to 64, the performance still is not improved. The

main reason is that the incoming bursty traffic is segment to multiple cells. Each cell will occupied one location of scheduling table. This seriously causes the HOL blocking happening in the FIFO address queue.

5 Conclusions

In this paper, we proposed an efficient multicast switch BRMSCS switch. The BRMSCS switch is organized by shared memory, crossbar fabric and BRMSCS. The BRMSCS scheduler uses larger tables to reducing HOL blocking happening inside the scheduler and guarantee the output order is memory access conflict free. We had simulated the BRMSCS under three different traffic models and also compare BRMSCS to MSCS. The simulation results show that BRMSCS performs better than MSCS under uniform and multicast random traffic. Under multicast bursty traffic, both BRMSCS and MSCS perform poor. However, the BRMSCS still perform better than MSCS.

References

1. Mark, J., Karol, et al.: Input Versus Output Queueing on a Space-Division Packet Switch. IEEE Transactions on communications 35(12), 1347–1356 (1987)
2. McKeown, N., Anderson, T.E.: A Quantitative Comparison of Scheduling Algorithms for Input-Queued Switches. Computer Networks and ISDN Systems 30(24), 2309–2326 (1998)
3. McKeown, N.: Fast Switched Backplane for a Gigabit Switched Router. Business Communications Review 27(12) (1997)
4. Chao, H.J.: Saturn: A Terabit Packet Switch Using Dual Round-Robin. IEEE Communications Magazine 38(12), 78–84 (2000)
5. Li, Y., Panwar, S., Chao, H.J.: The Dual Round-Robin Matching Switch with Exhaustive Service. In: IEEE Workshop on High Performance Switching and Routing, pp. 58–62 (May 2002)
6. Prabhakar, B., McKeown, N., Ahuja, R.: Multicast scheduling for input-queued switches. IEEE J. Select. Areas Commun. 15, 855–866 (1997)
7. Chen, W.-T., et al.: An Efficient Cell-Scheduling Algorithm for Multicast ATM Switching System. IEEE/ACM Transactions on Networking 8(8), 517–525 (2000)

Object-Based Video Coding for Distance Learning Using Stereo Cameras

Amir Hossein Khalili, Mojtaba Bagheri, and Shohreh Kasaei

Sharif University of Technology, Tehran, Iran
{A_Khaili,Mo_Bagheri}@ce.sharif,edu, skasaei@sharif.edu

Abstract. This paper presents a novel video encoding method for cooperative educational dissemination systems. Taking into consideration the inherent characteristics of stereo cameras framework in our educational videos and the ability of determining objects in different depths in a scene, we have proposed a novel object-based video encoding based on "sprite coding" that supports the MPEG-4 Version 1 Main profile in order to transfer distance learning videos across narrow-band transmission links such as the Internet. This paper proposes a multi-layer video object layer generation scheme with foreground moving object extraction and background sprite generation using stereo camera property. The foreground object is coded as a video object plane of its related layer, while the background sprite is coded using sprite coding in MPEG-4. We call this coding scheme "sprite mode". Experiments are conducted on video object plane generation and video coding using MPEG-4. We have compared the performance of our sprite mode with MPEG-4 normal mode and have shown that the coding efficiency of the sprite mode is higher than that of the normal mode at the same objective image quality when the foreground ratio is around 30%.

Keywords: Object-Based Video Coding, Distance Learning, Stereo Cameras, Motion Segmentation, MPEG-4.

1 Introduction

Advancements in communication technology are changing the way people around the world teach and learn. Since the Internet bulletin board systems (BBSs), email, and multimedia have already become parts of most college students' daily lives, applying these new communication technologies on instruction in technical communication is a great challenge for schools, teachers, and researchers in conventional classrooms as well as in distance learning environments. One of the main aspects of distance learning systems is educational video webcasting. Because of existence of limited and low bandwidth connections to the Internet in the developing countries (mostly 56 kb dial up modem connections), introducing new video encoding methods for real-time video coding is still a very important field of research [1].

Wide variety of methods has been reported to compress video streams, but a few of them have introduced context dependent methods. This paper describes an encoding/decoding model to compress educational video streams based on the "sprite coding" (the coding tool that supports the MPEG-4 Main Profile) and proposes a new

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 176–185, 2008.

method to automatically split a sequence into sprite and foreground objects using a stereo camera framework (where a sprite is defined as an image composed of pixels belonging to a video object visible throughout a video segment).

In our stereo camera framework, two cameras with slightly different viewpoints are mounted in front of the classroom so that cameras capture overlapped video sequences from the scene. We first define a model that consists of multilayer model of video object layers (VOLs); namely the foreground layers and the background sprite layer. Then, the layers automatically split into foreground video object planes (VOPs) and background VOPs using the stereo camera data. The multilayer VOP generation algorithm consists of a real-time background sprite generation method (with updating procedure) and a real-time foreground object extraction method. We show that the coding efficiency of our propose model is very desirable for real time distance learning video dissemination across narrow-band transmission links such as the Internet.

The rest of paper is organized as follows. In Section 2, a short review is given. The proposed algorithm are introduced in Section 3. The experimental results are discussed in Section 4 and finally, Section 5 concludes the paper.

2 Literature Review

In this section a short review on the related works is given.

2.1 Object-Based Coding in MPEG-4

When comes to "object-based" coding, the most significant feature of MPEG-4 is that it allows the separate encoding of foreground objects and background scene. According to [2], the scalable core profile uses the "binary shape" tool, which uses a constant non-zero value to represent the front object while padding all of the remaining pixels with "0". Compared with the rectangular-shaped MPEG-4 visual coding, the arbitrary-shaped coding is supposed to keep the quality of what concerns people the most while considerably cutting the bit budget. However, the arbitrary-shaped coding will include shape information in the compressed stream; while this additional overhead is not found in the rectangular-shaped codecs.

Fig. 1 presents the logic structure of an object-based MPEG-4 stream. As shown in this figure, a VO in MPEG-4 is described in terms of the associated closed caption text, if any, the kind of camera operation (fixed, panning, tracking, zooming, etc…) and its dynamic. The descriptors we chose for VOs derive from the analogy we pushed between the hierarchical representation based on video, VOs, and VOPs and the classical representation based on video, shots, and frames.

In the MPEG-4 framework, similar to a video frame in MPEG-2, a temporal instance of a video object is called a VOP [2]. In MPEG-4, the texture and the shape of each VOP are coded separately, where the texture coding of VOPs is similar to the coding of frames in MPEG-2 [2].

In MPEG-4, the shape of a VOP is described by a binary alpha plane, which indicates whether or not a pixel belongs to a VOP. Intra, Predicted and Bi-directionally predicted VOPs are the basic approaches to cope with arbitrary shaped images that differ from the conventional square images. The I- and P-VOPs can be used in the

"Simple Profile" whereas the B-VOP can be used in the "Core Profile". Also, the binary/grayscale alpha plane represents levels of transparency when two or more objects overlap. The binary alpha is used in the "Core Profile" whereas the grayscale alpha can be used in "Main Profile".

"Sprite" is one of the coding tools in the "Main Profile". The MPEG-4 standard was designed on the assumption that "Sprite" is provided in a certain way. How to generate the "Sprite" lies outside the scope of MPEG-4. In the next section, we focus on MPEG-4 "Sprite" coding used in combination with automatic "Sprite" generation.

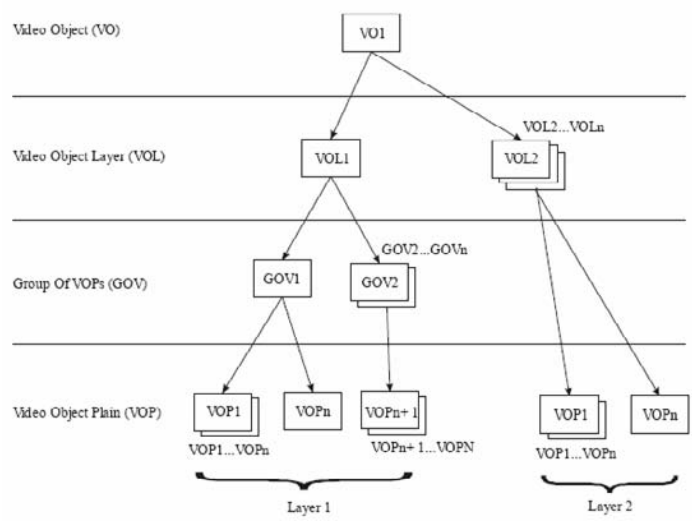


Fig. 1. Logic structure of an MPEG-4 stream

2.2 Sprite Coding in MPEG-4

Sprite coding expresses the area that occupies the same position across several frames as one plane (sprite). As such, the corresponding area in each frame can be regenerated from the sprite. If this area be large, most of the moving image region can be expressed using one sprite (static image), which can lead to a very high compression efficiency. MPEG-4 Version 1 Main Profile supports "sprite coding". But, the main challenge in this kind of scenario is actually generating the sprite itself. Fig. 2 shows the difference between conventional coding schemes and MPEG-4 object coding. The left hand side of this figure presents conventional coding in which a video sequence is coded frame by frame using motion compensated frames to reduce frame redundancy. The right hand side of this figure presents the typical MPEG-4 object coding process in which a video sequence can be divided into several video layers and objects; where one of the layers is the "Sprite". These video objects are assembled at the receiver side to reconstruct the image.

For automatic and real-time generation of sprites, Irani [4] *et al*. uses a top-down approach to calculate the global motion by aligning the entire image. Wang [5]

et al. generates multiple sprites from multiple global motions determined by the clustering of local motion. Lee [6] *et al.* report manual sprite generation and achieved dramatic compression efficiency while keeping the same subjective image quality. However, none of these methods have discussed the area outside the sprite (foreground area).

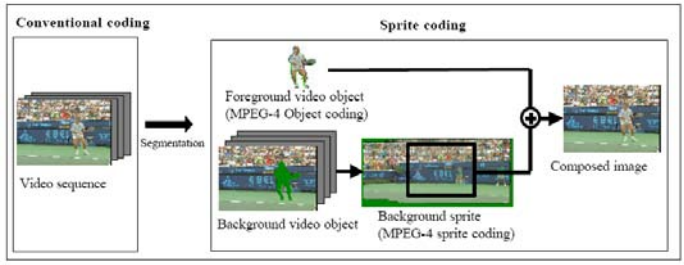


Fig. 2. "Sprite Coding" in MPEG-4 [7]

2.3 Foreground VOP Generation

How to express and code the non-background area (foreground area) is VOP generation algorithms in MPEG-4 object coding. If there are multiple objects within a frame, object coding is executed for each object. This means that each object must keep its correspondence across several frames. One of the conventional object extraction schemes (semi-automatic scheme) proposed by Choi *et al.* in [8] clearly solves the correspondence problem of each object, but its processing cost is not desirable. Some methods [10] extract and trace single objects. However, none of them discuss the correspondence of multiple objects across several frames.

3 Proposed Method

Fig .3 demonstrates an overview of our proposed method. Two cameras with slightly different field of views mounted in front of the classroom send their captured streams to encoder subsystem (See left hand side of Fig. 3.).

At encoder the captured video streams are decomposed to *video object layers* (VOLs) and *video object planes* (VOPs). Encoder side do these using three main components: "stereo matching", "automatic scene layering", "and sprite and VOPs generation" which will be explained in details. VOLs and VOPs are then multiplexed and coded using MPEG-4 standard "Sprite Coding" and then sent through network.

At decoder side, to reconstruct a full view of the scene, the decoder aligns all of the demultiplexed VOPs properly and blends them seamlessly using a composition algorithm. A wide variety of techniques has been reported for VOPs composition. In this paper, we have adopted the robust and efficient method proposed in [7].

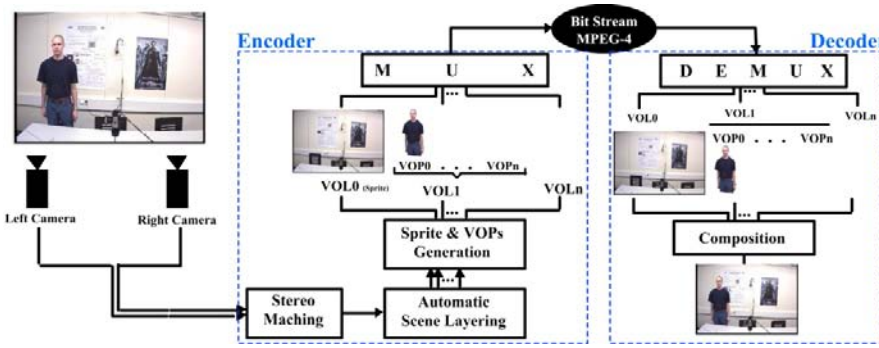


Fig. 3. Encoder/decoder model of our proposed system

3.1 Stereo Camera Frameworks

In our framework, two side-by-side cameras viewing the front scene of the classroom are employed. Fig.4.a shows a set-up of our system and Fig.4.b demonstrates its top view. In distance learning applications the arrangement of objects in a scene is so that the instructor occludes only some parts of the board. The distance between two cameras is chosen to the extent that if some part of the board gets occluded in one view, it can be observed in the other view. Thanks to the close side-by-side camera positioning, each camera takes a view of the same area from a slightly different angle. The two camera views have plenty in common, while each camera captures the visual information that is not in the view site of the other camera.

The algorithm determines the pixel correspondences between the two captured frames and produces a disparity map which indicates the relative offset between a pixel in the left image and its corresponding pixel in the right image. In fact, the disparity map contains the critical information needed to generate the VOLs.

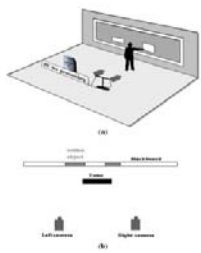


Fig. 4. a) Stereo cameras structure. b) Top view of the scene shown in (a).

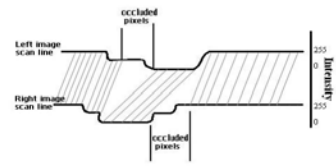


Fig. 5. Left and right scanlines. Matched pixels are shown by connecting lines. The pixels that can not be matched exactly create occluded regions.

3.2 Stereo Matching

We have adopted the stereo algorithm described in [10] which matches individual pixel intensities along scan lines, using a maximum likelihood cost function and

several cohesion constraints. The advantages of this approach are as follows. I) It provides a piecewise-constant and dense disparity map. II) The feature extraction and adaptive windowing techniques common in other stereo algorithms are avoided. III) It is fast and offers the potential for real-time implementation. IV) It handles large untextured regions which are common in our application.

3.3 Stereo-Based Automatic Scene Layering

Matched pixels construct correspondence sequence which is sequence of ordered pairs (x, y). x denotes the coordinate of a pixel in the left view and y denotes the coordinate of its corresponding point in the right view.

The disparity map $\partial(x)$ of a pixel x in the left scan line that matches a pixel y in the right scan line is defined as x - y, while the disparities of all pixels in an occluded area are assigned to the disparity of the neighboring points which are similar to them. The depth-discontinuity pixels are labeled as those pixels that border a change of at least two levels of disparity. In fact, by finding disparities we are determining the scene layers, Fig .5. We can deal with each layer as a separate VOL in MPEG-4 standard where any of its objects form a VOP in the related VOL.

Usually it is useful to construct piecewise-constant disparity maps. So that each object is assigned to a single disparity level, even if the depth of that object varies. Although this constraint sacrifices accurate scene reconstruction, but it facilitates working with each object as a VOP in a specific VOL.

In distance learning applications, we categorize the scene layers into two main categories: board layer (background sprite layer), foreground layer. We should track the instructor and send the information of the board frequently.

3.4 Sprite and VOPs Generation

As demonstrated in Fig .3, on the encoding side, the image is first split into multilayer video objects; namely the foreground object layers and the background sprite layer. The sprite layer consists of pixels belonging to a video object visible throughout a video segment as background layer. The foreground VOPs are any moving areas that do not belong to the background sprite; where each of these areas is treated as a VOP.

The foreground object and the background sprite are treated as independent VOPs, and each of them is converted to the free shape code in MPEG-4; the latter is subjected to enabling of “sprite coding” in the Main Profile of MPEG-4. These isolated bit streams are multiplexed and sent to the decoder. At the receiving side, the bit stream is demultiplexed and the VOPs are decoded, superimposed, and displayed.

Automatic multilayer VOP generation algorithm consists of 2 main steps: background sprite generation and foreground object extraction.

3.4.1 Background Sprite Generation

A video sequence usually contains of a background object (sprite) and many foreground objects. There are two steps involved in the generation of sprites namely initialization and updating. The Sprite is initialized from the very first VOP of the video

object sequence by just generating a background layer of the sequence as a sprite VOP but portions of this background image may not be visible in the initial frames due to the occlusion of foreground objects. In our proposed stereo camera framework, the sprite contains all parts of the background that were at least visible in one view, therefore occluded information in one view is registered accurately in the base view of the background sprite (see Fig. 6). This can be accomplished with assigning appropriate prospective transformations on occluded information in one view to map it to the base view sprite.

In the second step, the sprite is updated with each subsequent updating VOPs. This is achieved by estimating the sprite updating VOPs with respect to differentiation in the consequent constructed sprites. As a result, changes in the sprite VOL affect by generation of the sprite updating VOPs in "sprite mode" in MPEG-4 main profile.

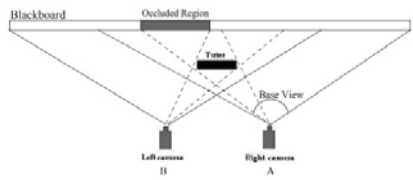


Fig. 6. Sprite generation in stereo frameworks

3.4.2 Foreground Object Extraction

The foreground object extraction step is based on segmentation and tracking of moving regions. In distance learning sequences the instructor VOP is treated as the main foreground object. As shown in Fig .7, in most scenarios of distance learning, the instructor initially exists in streams. This fact causes most background subtraction methods to segment foreground regions incorrectly. Fortunately, by using stereo cameras and knowing the fact that the instructor belongs to the nearest disparity level (with lots of motion) we can segment the related region concisely. To do so, we compute the optical flow of the frame and segment the nearest disparity level with considerable motion as the VOL of the instructor.

The segmented objects in each frame are converted to the free shape and are coded using MPEG-4 standard.

4 Experimental Results

In order to implement our proposed method, we have used OpenCV library and have run our code on the CPU 1.88 CoreDuo with 2 Giga Byte RAM. To give a complete performance comparison, we have used three standard stereo sequences, "Simon"," Jamie" and" Gide" [11].

4.1 Stereo Camera-Based Automatic VOP Generation

In order to evaluate the performance of our proposed method in VOP generation using stereo camera framework, we used three alternative methods that are mostly related to

the proposed method. These include most conventional background subtraction methods, such as temporal averaging model, gradient Gaussian model [12], and mixture of Gaussian model [13] for motion segmentation.

In our proposed method the moving object in the scene (instructor) is segmented as the base VOP. Fig. 7 and Fig. 8 present visual results over two sample videos of distance learning. To compare the segmentation quality of different methods, we have constructed a ground truth version of the frames and have found the quality of each method using the error measure. Fig .7 and Fig .8 shown how our method compensate false corrects and Table 1 lists the percentage of correct pixels found in each method and Table 2 shows the complexity cost of each method.

Fig. 7 and 8. (a) Frame No. 2, 5, and 44 of "Simon" and "Jamie" sequences. (b) Extracted foreground using background subtraction methods. (c) Disparity maps. (d) Extracted foreground using proposed method.

Table 1. Performance comparison of different VOP generation methods using error measure

Video Type	Video Size	Temporal Averaging Model	Gradient Gaussian Model	Mixture of Gaussian Model	Proposed Method
"Simon"	320×240	91.31%	93.46%	94.55%	95.22%
"Jamie"	320×240	90.99%	92.45%	94.27%	97.86%
"Gride"	640×480	88.44%	91.33%	93.67%	93.82%

Table 2. Complexity cost of different VOP generation methods (in milliseconds)

Video Type	Video Size	Temporal Averaging Model	Gradient Gaussian Model	Mixture of Gaussian Model	Proposed Method
"Simon"	320×240	1.21	162	198	109
"Jamie"	320×240	1.54	231	252	157
"Gride"	640×480	8.32	308	386	193

4.2 "Sprite Mode" in MPEG-4

As Table 3 shows, compressed video files when using "sprite mode" in MPEG-4 using our proposed method leads to higher coding efficiency when compared to the

Table 3. Compressed video size comparison in different video coding methods (in MB)

Video Type	Video Size	Original Video	H.264 Encoded	MPEG4 "Normal mode"	Proposed Method
"Simon"	320×240	3.847	1.023	1.048	0.785
"Jamie"	320×240	5.371	1.357	1.553	0.871
"Gride"	640×480	4.865	1.148	1.893	1.012

MPEG-4 "normal mode". Also, we have compared our results with H.264 AVC and have found that the proposed method leads to better results.

5 Conclusion

We overviewed the new video coding standard MPEG-4 main profile and object types. We then proposed a new video coding algorithm for distance learning video in stereo camera framework that well utilizes the characteristics of MPEG-4 visual tools such as sprite and video objects. The automatic multi-layer VOP generation algorithm was proposed and applied to low bit rate video coding. In simulations, the proposed algorithm was found to provide a dramatic increase in compression rates, 50% to 25%, compared with the normal MPEG-4 mode. It offers the same quality video when the foreground object size is around 30% of the image.

Acknowledgment

This work was in part supported by a grant from ITRC.

References

1. Bagheri, M., Lotfi, T., Darabi, A.A., Kasaei, S.: Content-Based Video Coding for Distance Learning. In: The 7th IEEE International Symposium on Signal Processing and Information Technology, ISSPIT, Cairo, Egypt (December 2007) (in press)
2. ISO/IEC 14496-2: 1999(E), Information Technology-Coding of audio-visual objects-Part 2: Visual (1999)
3. Farin, D., de With, P.H.N., Effelsberg, W.: Minimizing MPEG-4 Sprite Coding-Cost Using Multi-Sprites. In: SPIE Visual Communications and Image Processing, vol. 5308/1, pp. 234–245 (January 2004)
4. Irani, M., Hsu, S., Anandan, P.: Video Compression Using Mosaic Representation. Signal Processing: Image Communication 7(4-6), 529–552 (1995)
5. Wang, J., Adelsen, E.: Representing Moving Images with Layers. IEEE Trans. on IP 3(5), 625–638 (1994)
6. Lee, M., Chen, W., Lin, C., Gu, C., Markoc, T., Zabinsky, S., Szeliski, R.: A Layered Video Object Coding System Using Sprite and Affine Motion Model. IEEE Trans. on CSVT 7(1), 130–145 (1997)
7. Watanabe, H., Jinzenji, K.: Sprite coding in object-based video coding standard: MPEG-4. In: Proc. World Multiconf. On SCI 2001, XIII, pp. 420–425 (2001)

8. Choi, J.G., Lee, S., Kim, S.: Spatio-Temporal Video Segmentation Using a Joint Similarity Measure. IEEE Trans. on CSVT 7(2), 279–286 (1997)
9. Mech, R., Wollborn, M.: A Noise Robust Method for Segmentation of Moving Objects in Video Sequence. In: IEEE ICASSP 1997, pp. 2657–2660 (April 1997)
10. Birchfield, S., Tomasi, C.: Depth Discontinuities by Pixel-to-Pixel Stereo. International Journal of Computer Vision 35, 269–293 (1999)
11. http://research.microsoft.com/vision/cambridge/i2i/DSWeb.htm
12. Stauffer, C., Grimson, W.: Adaptive Background Mixture Models for Real-Time Tracking. In: Proc. Conf. Computer Vision and Pattern Recognition, vol. 2, pp. 246–252 (1999)

Seabed Image Texture Segmentation and Classification Based on Nonsubsampled Contourlet Transform

Reza Javidan[1], Mohammad A. Masnadi-Shirazi[2], and Zohreh Azimifar[3]

[1] Malek-Ashtar University of Technology, Shiraz, Iran
reza.javidan@gmail.com
[2] Dep. of Electrical Engineering, Shiraz University, Shiraz, Iran
masnadi@shirazu.ac.ir
[3] Dep. of Computer Science and Engineering, Shiraz University, Shiraz, Iran
azimifar@shirazu.ac.ir

Abstract. In this paper, a new split and merge algorithm based on nonsubsampled contourlet transform for automatic segmentation and classification of seafloor images is presented. This transform is a redundant version of contourlet transform which is a new two-dimensional extension of wavelet transform using multiscale and directional filter banks. It allows analysis of images at various scales as well as directions, which effectively capture smooth contours that are the dominant features in seabed images. The introduced redundancy brings simplicity and accuracy for feature calculation. The proposed method provides a fast tool with enough accuracy that can be implemented in a parallel structure for real-time processing. In addition, the simulation results are compared with the results of wavelet-based methods as well as other known techniques to show the effectiveness of the proposed algorithm.

Keywords: Seabed, contourlet transform, texture, segmentation, classification.

1 Introduction

The studies and technologies of underwater exploration can be used in various disciplines of underwater research. Today, sonar systems provide near-photographic images of underwater areas, even in zero visibility water [1],[2]. One of the most important applications of sonar systems is the automatic segmentation and classification of the sea bottom area [3]. Due to highly textured appearance of sonar images, texture analysis techniques are a common choice for seafloor acoustic images [4]. It was shown that Multiresolution space/scale methods such as wavelet transform are applicable for texture analysis [5]. One major advantage of wavelet analysis [6] is its ability to perform local analysis for revealing various aspects of data like trends, breakdown points, discontinuities in higher derivatives, and self-similarities. While seabed images like more natural images contain intrinsic geometrical structures that are key features in image analysis, one major drawback for two-dimensional wavelets is their limited capability in capturing directional information. To overcome this deficiency, researchers have recently come up with a new family of wavelet methods that can capture the intrinsic geometrical structures such as contourlet transform

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 186–193, 2008.

family [7]. Contourlet transform [8] overcomes directionality lack of 2-D wavelets by geometrically representing smoothness of contours. Contourlet transform allows having different and flexible number of directions at each scale, which makes it a suitable tool for seabed image analysis. In addition, the contourlet transform uses iterated filter banks which efficiently requires *O(N)* operations for an n-pixel image. The nonsubsampled contourlet transform [9] is a redundant version of contourlet transform which is shift invariant and brings more flexibility, simplicity and accuracy for feature calculation.

Common classification algorithms assign each pattern to one and only one cluster. This means that patterns are partitioned into disjoint sets. These algorithms perform well for compact and well-separated classes. However, in most realistic cases, especially in seabed classification, the distributions of two clusters usually overlap in feature space. The boundaries between different classes cannot be defined in specified directions in a clear-cut fashion. One reason is that the feature vector is computed over windows which might simultaneously cover different seafloor types, resulting a mixture of classes. Another reason is the ambiguity that exists between the definitions of for example, rocks and pebbles [10].

In this paper, a modified split and merge algorithm based on the concept of the nonsubsampled contourlet transform for segmentation and classification of the seabed images is presented. We also show that energies of the transformed coefficients are significant features for our application. We suggest solutions to improve texture segmentation quality using feature smoothing technique. The classification is done directly on the blocks of the image and the boundaries produced between known classes are refined to enhance the results. We aim to develop a fast algorithm with enough accuracy for real-time processing of textural images of huge size. The novelty of this approach is demonstrated by comparing the results with the results obtained in wavelet domain as well as other well-known methods.

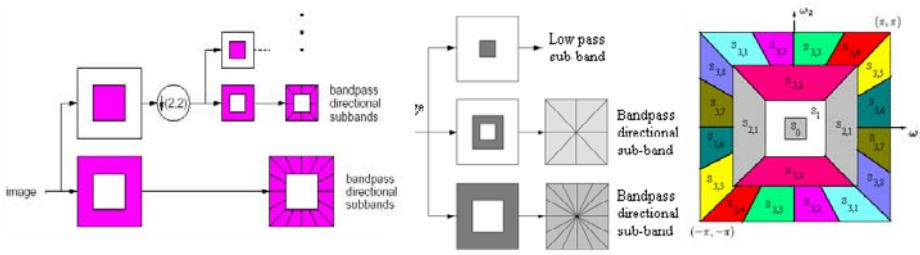
The content of this paper is organized as follows: In Section 2, the nonsubsampled contourlet transform is explained. Our new approach for automatic segmentation and classification of sea bottom based on the nonsubsampled contourlet transform is presented in Section 3. Finally experimental results are given and discussed in Section 4.

2 The Nonsubsampled Contourlet Transform

The contourlet transform is a new two-dimensional extension of the wavelet transform proposed by Do and Vetterli [8] using multiscale and directional filter banks with critical down-sampling operation. The contourlet expansion is composed of basis images oriented at various directions in multiple scales, with flexible aspect ratio that could effectively capture smooth contours of images. The contourlet transform, , as illustrated in Fig. 1(a), employs an efficient tree structured implementation, which is an iterated combination of the Laplacian Pyramid (LP), for capturing the point discontinuities, and the Directional Filter Bank (DFB), to gather the nearby basis functions and link point discontinuities into linear structures [8]. Since the DFB was designed to capture the high frequency directionality of the input image and it is poor on handling low frequency content, hence the DFB is combined with the LP, where the low frequencies of the input image is removed before applying DFB. It has been shown that

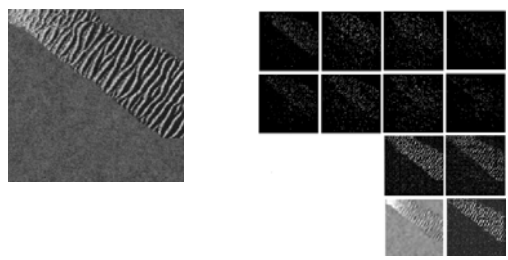
the discrete contourlet transform is shift variant, and achieves perfect reconstruction and has a redundancy ratio that is less than 4/3 [8].

The nonsubsampled contourlet transform [9] is a redundant expansion of the contourlet transform to allow practical processing on equal-sized directional sub-bands. The redundancy is achieved by discarding any down-sampling operation in the Laplacian pyramid scheme. Fig. 1 (b) displays an overview of the nonsubsampled contourlet transform while Fig. 1 (c) shows the frequency division of the nonsubsampled contourlet decomposition, where the whole spectrum is divided both angularly and radially and the number of directions is increased with frequency. Fig. 2 shows an example of a real seabed image and its corresponding three levels pyramidal nonsubsampled contourlet transform with 12 directional bandpass sub-bands.



(a) (b) (c)

Fig. 1. The block diagram of (a) standard contourlet transform. (b) nonsubsampled contourlet transform. (c) frequency partitioned for for 3 levels of the nonsubsampled contourlet transform.



(a) (b)

Fig. 2. (a) A real seabed image. (b) the result of 3 levels of nonsubsampled contourlet transform.

3 The Proposed Approach

Seabed images usually cover a large extent of area and may contain different types of textures. Therefore, they should be segmented into different classified regions. Fig. 3 illustrates block diagram of the proposed algorithm. Three levels nonsubsampled contourlet transform with 12 directional sub-bands are applied to the original image. The transform coefficients in different sub-bands are divided into blocks and energy value of each block is calculated to construct the feature vector. The estimated texture

features are then smoothed by a quadrant filtering method to reducing the variability of the estimates while retaining the region border accuracy. Euclidian distance classifier [12] is used for classification of each block. Finally, the classified blocks of the segmented image are refined and merged to produce the final segmented image.

3.1 Methodology

Following the block diagram of Fig. 3, the seabed image acquired from a sonar system is first transformed. In this research three levels of decomposition are considered using maxflat filters [9]. Three directional decompositions are performed in the lowest pyramid level, using dmaxflat7 filters [9]. Therefore, based on Fig. 1 (c), we obtain 12 directional bandpass sub-bands (8 at level 3, 2 at level 2, 1 at level 1, and 1 at level 0). Let $S_{0,0}$ be the lowpass sub-band, and $S_{1,1}$, $S_{2,1}$…$S_{2,2}$, and $S_{3,1}$…$S_{3,8}$ be bandpass directional sub-bands at the first, second, and third level, respectively.

A general pattern recognition paradigm achieves this task in two stages: first feature extraction and then, classification. We experimentally observed that local energy of the nonsubsampled contourlet coefficients provide good separated feature space. Since the energy of natural textures is mainly concentrated in the mid-frequencies, contourlet transform can preserve most of the original signal energy and can provide more reliable description of the texture [5]. Hence the distribution of energy can be selected as a valuable feature. The coefficients of all sub-bands are partitioned into non-overlapping blocks of size $M \times M$. The size M depends on the resolution of seabed textures. The finer the texture is, the smaller the size M should be selected. In this research and due to the type of our sonar images, M is chosen to be 4.

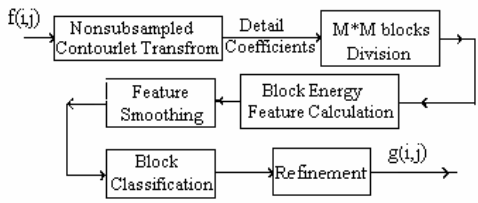


Fig. 3. Block diagram of the proposed method

Squared root of average energy of the nonsubsampled contourlet coefficients of the $M \times M$ block on each sub-band is calculated as:

$$E_{u,v}(x,y) = \sqrt{\frac{\sum_{i=x}^{x+M-1}\sum_{j=y}^{y+M-1} S_{u,v}{}^2(i,j)}{M^2}} \tag{1}$$

The process starts from top left corner of each sub-band and continuing in raster scan manner to form the feature vector $\underline{E}_{u,v}$ with size $N/M \times N/M$, where the original image is of size $N \times N$. For three levels of decomposition, the resulting nonsubsampled contourlet feature vector $\underline{\theta}$ consists of 12 energy feature vector, is given by:

$$\underline{\theta} = [\underline{E}_{0,0}, \underline{E}_{,11}, \underline{E}_{2,1} ... \underline{E}_{2,2}, \underline{E}_{3,1} ..., \underline{E}_{3,8}] \tag{2}$$

As mentioned above, in seabed classification, since the feature vector is computed over blocks which might simultaneously cover different seafloor types, the distributions of two clusters usually overlap in feature space and the boundaries between different classes cannot be defined clearly. Moreover, there is an ambiguity between the definitions of different seabed materials. To avoid such problems, we can perceive the process of calculating the local energy as the problem of smoothing a noisy image. In noise smoothing, we always face the problem of how to smooth noise without blurring the edges. In texture features, our problem can be formulated as how to estimate texture features from the feature image $\underline{E}_{u,v}$ with less possibility of mixture of energy along region boundaries [11].

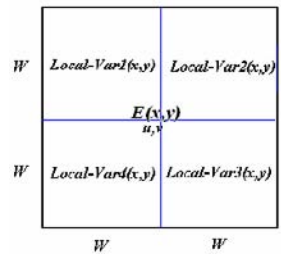


Fig. 4. Feature smoothing: quadrant windows of size $w \times w$ around feature point $E_{u,v}(x, y)$

Suppose $\underline{E}_{u,v}$, as shown in Fig. 4, is the feature estimated by (1). At each point $E_{u,v}(x, y)$, a set of four immediate neighborhoods of size $w \times w$ that lie on various sides of that point are examined. Then local sample variance, $Local\_Var(x, y)$, as the measure of variability over each neighborhood is used, and the pixel at point (*x,y)* is replaced by the average of the neighborhood that has the lowest variance; since those windows that contain region borders generally have a higher variability introduced by the edges. In this paper, *w=3* is selected as the size of local neighborhood. The process of feature smoothing is repeated for all the elements of the feature vector $\underline{\theta}$.

Euclidean distance [12] of smoothed feature block values is used for the block classification. Our experiments show that using other types of distance measurements did not improve the result much better, except increasing the simulation time. If $\underline{\theta}_p$ and $\underline{\theta}_q$ are the feature vectors of a given image under classification (*p)*, and a known image class (*q)*, respectively, the Euclidean distance between them is given by:

$$d_{Euclidean}(\underline{\theta}_p, \underline{\theta}_q) = \sqrt{\frac{1}{r}\sum_{u,v}(E^p{}_{u,v} - E^q{}_{u,v})^2} \tag{3}$$

where r=12 is the number of feature values and $E^p{}_{u,v}$ and $E^q{}_{u,v}$ denote the corresponding feature vector values for image *p* and *q* respectively. All blocks of the input image are classified according to the (3). After block classification, the input image is segmented block-wise, which is called “crisp segmentation”.

3.2 Boundary Refinement

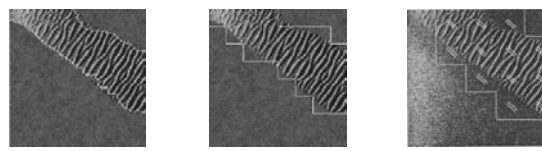
The refinement process on the classified regions obtained by $M \times M$ blocks of the input image includes the following steps:

I. **Block refinement:** Consider each $M \times M$ non-overlapping block in the segmented image. If the block's class label is the same with all 4-connected block neighbors, do nothing. Otherwise, if all four block neighbors or at least three of them have the same class labels similar to each other, change the class label of centered block as the class label of its major neighbors. This process should be done in a raster scan form, from top left to the right bottom of the image.

II. **Deleting disconnected islands:** The isolated regions with small area should be deleted from the segmented image obtained in previous part. Discard any disconnected region which has area less than a pre-defined threshold by combining it with its neighbors. This threshold depends on the sonar image resolution; the image with higher resolution, the larger threshold value should be selected.

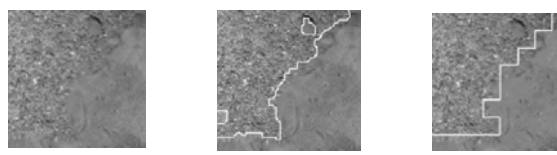
4 Experimental Results

To validate our method, a sample image shown in Fig. 2 is used and the results are shown in Fig. 5 (a) and 5 (b) show the segmentation results using energy features of the nonsubsampled contourlet coefficients with Euclidean distance, and the energy feature of the redundant wavelet transform coefficients with Euclidean distance classifier respectively.



(a) (b) (c)

Fig. 5. Segmentation results of Fig. 2 (a) using proposed method. (b) using redundant wavelet transform. (c) using Mignotte method [10].

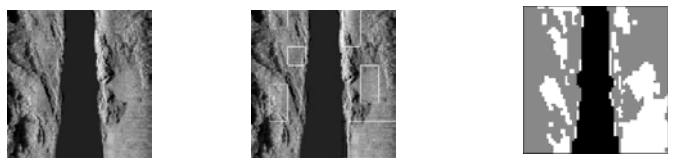


(a) (b) (c)

Fig. 6. (a) A real seabed image composed of gravel (left side) and rippled muddy sand (right side of the image) (b) the segmentation result using nonsubsampled contourlet domain features. (c) The segmentation result using redundant wavelet transform features.

(a) (b) (c) (d) (e)

Fig. 7. (a) Original seabed image composed of boulders (left) and sand (right side of the image) (b) the segmentation result using the proposed method based on nonsubsampled contourlet transform. (c) the result after refinement process (d) the final superimposed segmented image. (e) the segmentation result using 2 levels of wavelet transform with energy features and Euclidean distance classifier.



(a) (b) (c)

Fig. 8. A real side scan sonar image: (a) original image. (b) the segmentation result using 2 levels of standard wavelet transform with energy features and Euclidean distance classifier. (c) the segmentation result (shown with different gray level values) using energy features of the contourlet transform with Euclidean distance classifier and after refinement.

Mignotte and Collet [10] proposed a method for classification of seabed using high resolution sidescan sonar images. Their method consists of four stages: extracting shadows, shape parameter vectors computation, classification with a fuzzy classifier, and finally, refining the result using Markov random field model. For comparison, their result for the same seabed image is shown in Fig. 5 (c). It can be seen that our proposed segmentation method, is more accurate from perceptual point of view.

Another real seabed image is shown in Fig. 6 (a) which composed of gravel (left side) and rippled muddy sand (right side of the image). The segmentation results using nonsubsampled contourlet domain features and redundant wavelet transform features are shown in Fig. 6 (b) and Fig. 5 (c) respectively.

To understand the role of the refinement process, a seabed image composed of boulders (left side of the image) and sand (right side of the image) and its intermediate results of segmentation process are shown in Fig. 7. Fig. 7 (a) is the original image while Fig. 7 (b) is the segmentation result using the proposed method based on nonsubsampled contourlet transform and before the refinement process. Fig. 7 (c) shows the result after the refinement process and Fig. 8 (d) is the final superimposed segmented image. For comparison, Fig. 7 (e) is the segmentation result for the same image using 2 levels of standard pyramidal wavelet transform with energy features of the wavelet coefficients and Euclidean distance classifier. It is clear that the results obtained using contourlet transform are superior than that of the wavelet transform.

As one more example, Fig. 8 (a) shows a real side scan sonar image. The segmentation result using two levels of standard wavelet transform and energy feature of the

wavelet coefficients with Euclidance distance classifier is shown in Fig. 8 (b). The segmentation result using energy features of the nonsubsampled contourlet coefficients and after the refinement process is shown in Fig. 8 (c). As it is clear, the segmentation result obtained in contourlet domain is more accurate than that of the wavelet domain, while the same feature set and distance metric are used.

The structure of the proposed algorithm as shown in Fig. 1 is parallel and it can be used in a real time acoustic ground discrimination system. However, the algorithm is fast enough to be implemented on a personnel computer (PC). In a typical PC with 3 GHz AMD 64 CPU and 1 GB RAM and 128 MB PCIX graphic card, a block of size 32×32 will be classified in less than 0.0104 seconds and the average elapsed time for segmentation of an image of size 512×512 is 0.312418 seconds.

References

1. Javidan, R., Jones, I. S. F.: High Resolution Acoustic Imaging of Archaeological Artifacts in Fluid Mud. In: International congress on the application of recent advances in underwater detection and survey techniques to underwater archeology, Turkey (2004).
2. Javidan, R., Eghbali, H.J.: Seabed Textural Image restoration and Noise Removal Using Genetic Programming. In: 5th International Conference on Marine Researches and Transportation, Italy (2005)
3. Javidan, R., Eghbali, H.J.: Automatic Seabed Texture Segmentation and Classification Based on Wavelet Transform and Fuzzy Approach. International Journal of the Society for Underwater Technology 27(2), 51–55 (2007)
4. Tang, X.: Optical and sonar image classification: wavelet packet transform vs. Fourier transform. Computer Vision and Image Understanding 79, 25–46 (2000)
5. Arivazhagan, S., Ganesan, L.: Texture Classification Using Wavelet Transform. Pattern Recognition Letters 24, 1513–1521, 3197–3203 (2003)
6. Mallat, S.: A wavelet Tour of Signal Processing, 2nd edn. Academic Press, London (1999)
7. Po, D.D.-Y., Do, M.N.: Directional Multiscale Modeling of Images using the Contourlet Transform. IEEE Transaction on Image Processing 15(6), 1610–1620 (2006)
8. Do, M.N., Vetterli, M.: The contourlet transform: an efficient directional multiresolution image representation. IEEE Transactions on Image Processing 14(12), 2091–2106 (2005)
9. Cunha, A.L., Zhou, J., Do, M.N.: The Nonsubsampled Contourlet Transform: Theory, Design, and Applications. IEEE Transaction on Image Processing (2005)
10. Mignotte, M., Collet, C., Perez, P., Bouthemy, P.: Markov Random Field and Fuzzy Logic Modeling in Sonar Imagery: Application to the classification of underwater floor. Computer Vision and Image Understanding 79, 4–24 (2000)
11. Song, X., Chen, Z., Wen, C., Ge, Q.: Wavelet Transform-based Texture Segmentation Using Feature Smoothing. In: Proceedings of the Second International Conference on Machine Leaning and Cybernetics, Xi'an, November 2-5 (2003)
12. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. John Wiley and Sons, Inc., Chichester (1998)

Unequal Error Protection for the Scalable Extension of H.264/AVC Using Genetic Algorithm

Amir Naghdinezhad, Mahmoud Reza Hashemi, and Omid Fatemi

Nanoelectronics Center of Excellence, School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran
a.naghdinezhad@ece.ut.ac.ir, rhashemi@ut.ac.ir, omid@fatemi.net

Abstract. With the rapid development of multimedia technology, video transmission over error prone channels is becoming possible. Lossy video coding together with channel errors result in more degradation in quality of the video. Unequal Error Protection (UEP) can be employed for solving this issue. Using UEP with scalable video coding (SVC) improves the quality of the transmitted video. The efficiency of this combination can be further enhanced by carefully considering the importance of each protected element. In this paper, we propose an efficient UEP scheme to protect scalable video, coded with the scalable extension of H.264/AVC, over networks with packet loss. Genetic algorithm (GA) is exploited to achieve the optimal rate allocation for each layer. Experimental results show a significant improvement of 1.15dB in average, in comparison with conventional methods.

Keywords: Unequal error protection (UEP), scalable video coding (SVC), H.264/AVC, Genetic Algorithm (GA), Reed Solomon coding.

1 Introduction

In advanced multimedia services such as video phone, video conferencing and video streaming, compressed video signals are transmitted over error prone channels. In these channels, in addition to a limited channel bandwidth, packet loss is a main challenge. The limited bandwidth is especially a concern when multiple clients with different capabilities in terms of bandwidth, power consumption and display resolution simultaneously access the same compressed content. To address the limited bandwidth we need a better compression performance which results in lossy compression. Lossy video coding together with packet loss results in an even lower quality of service (QoS).

Scalable video coding (SVC) methods such as the scalable extension of H.264/AVC [1] can be employed for solving these challenges in error prone networks. By using scalable video coding, an efficient flexible bit stream is produced which alleviates the effect of network and application heterogeneity. This single bit stream contains the information to fulfill the requirements of different clients. Furthermore, since the video layers in an SVC stream have different importance, unequal error protection (UEP) can be applied on the video signal. Applying UEP on a scalable video signal improves the efficiency and reliability of the network.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 194–202, 2008.

Applying UEP on scalable video has been addressed by many researches with various types of scalability. In [2], the impact of applying UEP on different layers of scalable video has been studied. It allocates different amount of protection to base and enhancement layers of fine granular scalability (FGS) coding. UEP schemes that consider the different importance of I, P and B frames for MPEG-2/H.263 video have been also proposed in [3], and [4]. In [5], the two above aspects are jointly considered for channel rate allocation. It makes use of the different importance that layers and frames in an MPEG-2/H.263 video normally have. [6] addresses jointly considering the importance of both layers and frames for the scalable extension of H.264/AVC. However, when the number of layers and frames increase, the computational complexity increases dramatically. As a result, finding the proper channel rate in a reasonable time can be a challenge. In some works optimization techniques such as evolutionary algorithms have been used for this purpose [3], [5].

In this paper, an efficient UEP method is proposed that protects scalable video coded with the scalable extension of H.264/AVC using the genetic algorithm (GA). Genetic algorithm is a stochastic and population-based algorithm which has been successfully employed in many research and optimization problems [7]. The proposed algorithm considers the importance of each layer together with each frame. GA is exploited to achieve the optimal rate allocation for each layer. The proposed method makes use of Forward Error Correction (FEC) schemes based on Reed Solomon codes for unequal error protection.

The rest of this paper is organized as follows. In Section 2 a brief introduction on scalable extension of H.264/AVC is presented. Section 3 explains the proposed system architecture. Section 4 describes the proposed channel rate allocation scheme using GA. Finally simulation results are presented in Section 5 followed by conclusions.

2 Review of Scalable Extension of H.264/AVC

Since heterogeneous environments, where clients have different capabilities in terms of complexity, bandwidth, power and display resolution, are common in many multimedia applications, scalable video concept was introduced. By using scalable video coding, a flexible bit stream is produced which accommodates almost all the clients. SVC has been a research topic for more than 20 years, but previous scalable video coding standards have not been very popular. This was mostly because of the significant loss in performance, and the complexity of the decoding process introduced by the scalable video coding. In January 2005 MPEG and the ITU-T Video Coding Experts Group (VCEG) decided to jointly finalize the SVC project as an amendment of the H.264/MPEG4-AVC standard [1].

By using a hierarchical prediction structure, as illustrated in Fig., temporal scalable bit streams can be generated without changing the H.264/AVC parts. Key frames are intra coded or can be coded using the previous key frames. The other frames between any two key frames are hierarchically predicted. Fig. shows the dependencies between frames. The base layer consists of the key pictures, while the remaining frames are parts of the enhancement layers. For example, in Fig. 1 the base layer contains picture number 8. Picture number 4 builds the first enhancement layer. Pictures numbers 2 and 6 are part of the second enhancement layer and finally the remaining pictures

make the last temporal enhancement layer. All the pictures between two key pictures are called a group of pictures (GOP). As it is shown in Fig., there is also one key picture in each GOP.

Two kinds of signal to noise ratio (SNR) scalability or quality scalability are supported in the scalable extension of H.264/AVC. The base of coarse grain scalability (CGS) is changing the quantization parameters for different layers. These parameters are the maximum for the base layer coding and for each enhancement layers they decrease. As CGS has weaknesses, a fine granular scalability (FGS) has been introduced. In FGS, the base layer quality can be improved by progressive refinement (PR) slices. Each of these slices corresponds to a refinement of the residual data. The residual data are coded such that they can be decoded by applying just a single inverse transform for each block. By using progressive slices, the related NAL units can be truncated at different points. In Fig., three quality levels are shown for each temporal layer.

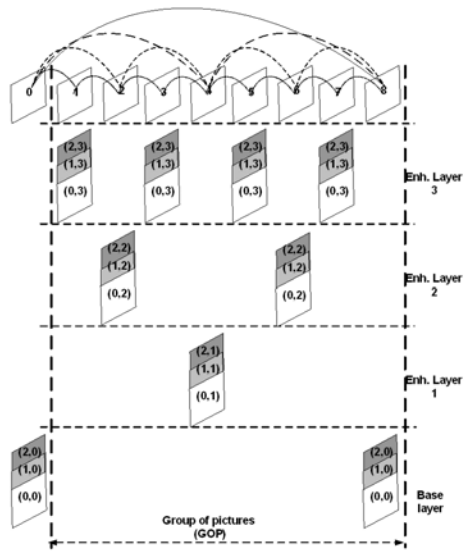


Fig. 1. Hierarchical prediction structure with 4 dyadic temporal and 3 quality levels. (Temporal, Quality).

3 Channel System Architecture

The proposed method makes use of Forward Error Correction (FEC) schemes based on Reed Solomon codes for unequal error protection. Reed Solomon codes are a type of linear non binary block codes. They result in maximum erasure protection while adding the minimum redundancy. A (n, k) Reed Solomon code is able to protect data against *n-k* symbol erasures. *k* is the number of data symbols and *n-k* symbols are added as parity symbols. *n* is the total number of symbols in a coded block. Each symbol is m bits where $n = 2^m - 1$. In this paper, we use Galois field (2^8) where the symbols are represented in bytes and n is equal to 255 [8].

Fig. shows the selected packetization scheme. By using this scheme, each layer can be protected independently. Each row corresponds to one layer with a specific temporal-quality resolution and each column represents one packet. The total number of packets is *N* and the packet size is set to *M* bytes. Furthermore, Reed Solomon coding is applied horizontally.

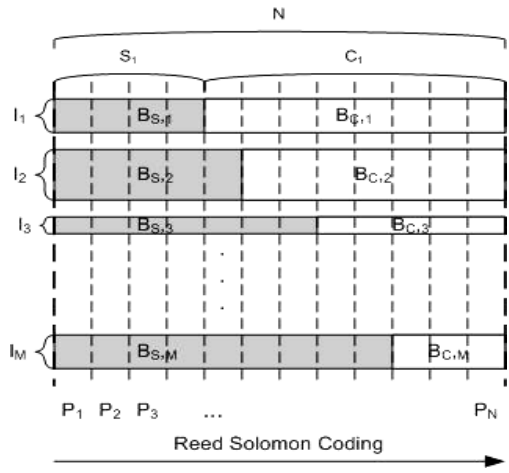


Fig. 2. Packetization scheme for unequal error protection

The *i*-th layer with $B_{s,i}$ bits source data is protected with $B_{c,i}$ parity bits. $B_{s,i}$ and $B_{c,i}$ are respectively distributed into S_i and C_i packets. The probability of successfully receiving the *i*-th layer with RS(N, C_i) along a network with packet loss probability p is:

$$q_i = \sum_{j=0}^{N-S_i} \binom{N}{j} p^j \times (1-p)^{N-j} \tag{1}$$

4 Channel Rate Allocation Method

The proposed method in this paper determines the amount of channel codes for each layer ($B_{c,i}$). Given a total bit budget, R_{total}, our method allocates channel bits such that the quality (Q_r) of the received video is maximized:

$$\text{Maximize } Q_r \text{ While } R_s + R_c \le R_{total} \tag{2}$$

The overall bit budget consists of overall source bits and overall channel bits. As shown in (3) and (4), total source bits, R_s, and total channel bits, R_c, are made up of $B_{s,i}$ and $B_{c,i}$ for i between *1* and *M*, respectively.

$$R_s = \sum_{i=1}^{M} B_{s,i} = \sum_{i=1}^{M} (S_i \times l_i) \tag{3}$$

$$R_c = \sum_{i=1}^{M} B_{c,i} = \sum_{i=1}^{M} (C_i \times l_i) \tag{4}$$

The Q_r of the received video is calculated based on the quality improvement of each received layer (5). $Q_{s,i}$ is the quality improvement of each layer at the encoding time. $Q_{s,i}$ is calculated based on the difference of PNSR of two coded stream, one with and the other without the i-th layer. q_i is the probability of successfully receiving the i-th layer and is calculated as (1).

$$Q_r = \sum_{i=1}^{M} Q_{r,i} = \sum_{i=1}^{M} (Q_{s,i} \times q_i) \tag{5}$$

To apply GA [7] to channel rate allocation problem, each gene characterizes a set of channel rates. In this experiment, a binary-encoded version of GA [9] is used to obtain acceptable channel rates. Therefore, the search space consists of sets of multi-dimensional channel rates. If we use l bit for each channel rate, the length of a particle will be $l \times n$, where n is the number of layers of different frames in each GOP. To evaluate the fitness of each particle in the population, we use (5) while considering the total bit budget constraint (2). The flowchart of the proposed channel rate allocation scheme using the genetic algorithm is shown in Fig..

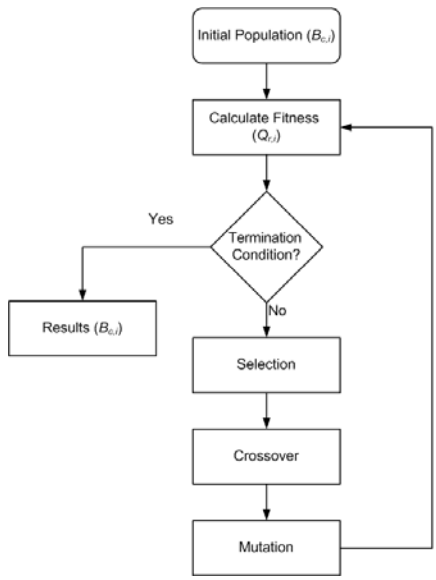


Fig. 3. Flowchart for the proposed channel rate allocation method

To determine the proper population size, various tests with populations size from 20 to 500 were performed. Experimental results indicate that although using 500 as the population size will increase the running time significantly, it does not improve

the fitness notably. Hence the size 200 was selected. In addition, simulation results show the 100 generations is enough for achieving optimum results. We also set the crossover and mutation probabilities to 0.8 and 0.02 respectively.

5 Simulation Results

In order to simulate a network that is subject to packet losses we employed a two-state Markov channel [10]. The average packet loss rates of 3%, 5%, 10%, 20% and 30% were tested. Since a two-state Markov channel has a random behavior, each simulation was performed 10 times and the average results are reported.

Standard sequences like "Football", "Foreman" and "News" with CIF format and "Crew" and "Carphone" sequences with QCIF format were used in these simulations. At the decoder side, if a frame is lost, the last correctly received picture is repeated in its place. This helps in calculating the PSNR. Joint Scalable Video Model (JSVM 7.0) [11] was used as encoder and decoder of scalable extension of H.264/AVC. The GoP size was set to 16 and one frame was intra coded in each 32 frames. The simulations were performed on 10 GoPs. In addition, adaptive inter layer prediction was enabled.

The channel rate of each layer is defined as the ratio of source bits ($B_{s,i}$) to sum of source bits ($B_{s,i}$) and channel bits ($B_{c,i}$):

$$Channel \quad Rate = \frac{B_{s,i}}{B_{s,i} + B_{c,i}} \times 100\% \tag{6}$$

Table 1. Channel rate allocation of (Temporal, Quality) layers at different packet loss rates for "Football" sequence with CIF format

(T,Q)	Packet Loss Rate				
	3%	5%	10%	20%	30%
(0,0)	87.00%	84.00%	77.00%	64.00%	51.00%
(0,1)	91.80%	92.00%	85.80%	85.20%	96.40%
(0,2)	92.00%	91.80%	86.10%	97.50%	100.00%
(1,0)	88.51%	91.20%	85.00%	72.70%	61.40%
(1,1)	88.87%	91.60%	85.20%	79.60%	92.50%
(1,2)	91.93%	92.20%	85.90%	100.00%	100.00%
(2,0)	89.69%	91.30%	85.00%	73.30%	63.70%
(2,1)	92.34%	92.65%	85.40%	77.10%	92.30%
(2,2)	91.46%	92.15%	88.90%	96.20%	98.15%
(3,0)	88.91%	92.28%	85.13%	73.68%	70.03%
(3,1)	88.83%	92.60%	86.70%	80.73%	94.40%
(3,2)	89.24%	92.40%	92.48%	98.75%	100.00%

In Table 1, the channel rate allocation of each temporal-quality layer is presented for different packet loss rates, for the "Football" sequence with CIF format. The channel rates for important temporal-quality layers are less. This means that more channel bits are allocated to these layers to protect them more. In addition, when the packet loss rate increases, the parity bits required to protect a part increase as well. As a result, when the bit budget limitation is not a concern, the channel rate will decrease for higher packet loss rates. But as the total budget is limited in this case, there are no channel bits available to protect less important layers. Hence, no parity bits are allocated to less important layers. Consequently, the channel rate of these layers will increase.

Table 2. Average PSNR comparison of the proposed method with other channel allocation techniques

Sequence	Packet Loss Rate	EEP (dB)	UEP1 (dB)	UEP2 (dB)	UEP3 (dB)	Proposed (dB)
Football	3%	35.27	35.27	35.27	35.27	35.27
	5%	34.61	34.94	34.99	34.94	34.97
	10%	32.44	33.89	33.67	33.70	34.12
	20%	22.35	26.92	28.48	29.46	30.96
	30%	19.05	23.06	25.88	26.83	28.34
News	3%	38.81	38.95	38.80	39.20	39.24
	5%	37.47	37.62	36.86	37.95	38.44
	10%	31.17	32.16	32.41	32.24	34.41
	20%	28.07	30.99	30.62	30.89	31.65
	30%	28.07	29.32	29.51	29.57	30.79
Foreman	3%	36.01	35.75	35.68	35.75	35.77
	5%	34.36	34.08	34.26	34.20	34.49
	10%	31.03	32.02	31.71	32.10	33.13
	20%	23.19	29.38	29.13	30.12	30.99
	30%	18.87	24.91	24.25	26.50	27.78
Carphone	3%	38.02	38.03	37.69	38.00	38.06
	5%	37.79	37.62	37.53	37.57	37.84
	10%	34.19	35.38	35.06	36.14	36.49
	20%	27.78	31.26	31.05	32.95	33.23
	30%	27.78	29.46	29.70	29.63	30.75
Crew	3%	36.11	36.11	35.88	36.11	36.11
	5%	35.51	35.69	35.65	35.37	35.62
	10%	32.42	34.59	34.30	34.64	35.00
	20%	27.83	31.26	30.68	30.83	32.11
	30%	24.02	27.78	28.05	29.44	30.13

In Table 2, the average PSNR of the proposed method is compared with four other channel allocation methods for five different sequences. "Football", "News" and "Foreman" sequences with CIF format and "Crew" and "Carphone" sequences with QCIF format are used in this test. The Equal error protection (EEP) method protects all the parts equally. The second compared method is UEP1, where UEP is applied

according to the different importance of each frame [3] and [4]. In the UEP2 method, UEP is applied on different quality layers [2]. Since [2], [3] and [4] have used MPEG-2 as their encoder and decoder and the results were not available for the scalable extension of H.264/AVC, we have implemented them ourselves by using the same GA algorithm, as in the proposed method. Although the implementation results may be slightly different from the original version, but they provide a good indication of the performance of the proposed method. The UEP3 method is similar to the proposed algorithm. It optimally performs the channel rate allocation by considering the importance of different layers and different frames, using an optimization technique other than the GA algorithm [6].

Our proposed method achieves an improvement of 0.65dB in average in comparison with the UEP1 method which uses a similar optimized method, as the proposed technique in this paper, except that it is not using the GA algorithm as its optimization method. In comparison with conventional UEP methods, the average improvement is about 1.15dB. In addition comparing the result with the EEP method shows an improvement of 2.49dB in average.

6 Conclusions

In this paper, we proposed an efficient UEP method to protect scalable video over error prone networks. Genetic algorithm (GA) was used to find the optimal allocation for each quality and temporal layer. The scalable extension of H.264/AVC is chosen as the encoder and decoder. Experimental results showed a significant improvement of 1.15dB in average, in comparison with conventional UEP methods.

Acknowledgments. The authors would like to express their gratitude to Iran Telecommunication Research Center (ITRC) for their support during this research.

References

1. Joint Video Team of ITU-T VCEG and ISO/IEC MPEG, Scalable Video Coding – Working Draft 1, Joint Video Team, Document JVT-N020 (January 2005)
2. van der Schaar, M., Radha, H.: Unequal packet loss resilience for fine-granular-scalability video. IEEE Trans. on Multimedia 3(4), 381–393 (2001)
3. Fang, T., Chau, L.P.: A novel unequal error protection approach for error resilient video transmission. In: Proc. IEEE Intl. Symp. on Circuits and Systems, May 2005, pp. 4022–4025 (2005)
4. Huang, C., Liang, S.: Unequal error protection for MPEG-2 video transmission over wireless channels. IEEE Trans. On Signal Processing Image Communication 19, 67–79 (2004)
5. Fang, T., Chau, L.P.: GOP-based channel rate allocation using genetic algorithm for scalable video streaming over error-prone networks. IEEE Trans. on Image Processing 15(6), 1323–1330 (2006)
6. Naghdinezhad, A., Hashemi, M.R., Fatemi, O.: A Novel Adaptive Unequal Error Protection Method for Scalable Video over Wireless Networks. In: Proceedings of the 11st International Symposium on Consumer Electronics (ISCE), Dallas, Texas, USA (June 2007)

7. Goldberg, D.E.: Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, Reading (1988)
8. Lin, S., Costello, D.J.: Error Control Coding: Fundamentals and Applications. Prentice-Hall, New York (1983)
9. Cormen, T.H., Leirson, C.E., Rivest, R.L.: Introduction to Algorithms. MIT Press, Cambridge (1990)
10. Elliott, E.O.: A model of the switched telephone network for data communications. Bell System Technical Journal 44(1), 89–109 (1965)
11. Reichel, J., Schwarz, H., Wien, M. (eds.): Joint Scalable Video Model JSVM-7. Joint Video Team, Doc. JVT-T202, Klagenfurt, Austria (July 2006)

An Adaptive Method for Moving Object Blending in Dynamic Mosaicing

Mojtaba Bagheri, Tayebeh Lotfi, and Shohreh Kasaei

Sharif University of Technology, Tehran, Iran
{Mo_Bagheri,T_Lotfi}@ce.sharif.edu, skasaei@sharif.edu

Abstract. Various video applications such as mosaicing and object insertion require blending of image regions. The blending quality is often measured subjectively by considering the similarity of the blended image to each of the input images and the visibility of the seams among stitched regions. This paper presents a novel method to blend the moving objects in the related background mosaic based on an adaptive alpha blending method. In order to compute the best possible value for the alpha blending coefficient in dynamic mosaic updating, here we propose a fuzzy method based on relative speed and scale parameters of moving objects. Conducted experiments show that the proposed method improves the quality of alpha blending method for real-time applications.

Keywords: Image blending, dynamic mosaicing, fuzzy, sprits, image stitching.

1 Introduction

Dynamic mosaicing is the process of reconstructing a wide scene model by aligning and properly blending together partially overlapped images acquired by a video sequence captured from a wide scene. In order to create a dynamic mosaic from captured sequence, we need an efficient algorithm to blend the moving objects in the static background mosaic (sprite) with minimum obstructive boundaries around overlapped regions and with smooth transitions in moving objects boundaries [1,2,3].

Wide variety of methods has been reported to blend and stitch objects in the scene, but none of them introduce suitable trade off between time complexity and visual quality. In this paper a novel method for blending moving objects in static background mosaic (sprite) has been introduced. This proposed method generates an alpha blending coefficient with respect to relative speed and scale of moving objects adaptively.

The rest of paper is organized as follows. In Section 2, some related literature on video mosaicing is given. A short literature review of different methods in object blending is introduced in Section 3. Different steps of the proposed algorithm are introduced in Section 4. The experimental results are discussed in Section 5 and finally, Section 6 concludes the paper.

1.1 Video Mosaicing

Video mosaicing is the process of constructing a wide scene model by aligning and properly blending the partially overlapped input images acquired by an image sequence captured from a wide scene.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 203–210, 2008.

There are two approaches in video mosaicing methods, static and dynamic. The static mosaicing methods operate on the batch mode by aligning all images in a fixed coordinate system. However, it cannot completely depict the dynamic aspects of an image sequence. The dynamic mosaicing method can overcome this problem because the content of each new mosaic scene model is updated with the most current information obtained from the most recent image [5,6].

1.1.1 Static Approach

In this method, each pixel of the created mosaic image M_t is computed using a linear combination of all available original background images as follows:

$$M_t(p) = \sum_{i=1}^{t} \beta_t^i(p)\tilde{I}_i(p) \tag{1}$$

Where, for each pixel P, and the pixel values are considered to be in the range [0-256], where the value 0 means that the corresponding pixel is undefined. The problem consists not only in estimation of the images for each original image, but also in determination of optimal values of each coefficient.

1.1.2 Dynamic Approach

In this method, each mosaic image Mt is iteratively updated using the previously computed mosaic image and the current image. Mathematically, the new mosaic image Mt can be calculated using the following equation, for which the first term corresponds to the intersection area between images Mt-1 and, and the second term to the rest of the mosaic image

$$M_t(p) = [\alpha_t\tilde{I}_t(p) + (1-\alpha_t)M_{t-1}(p)]\overline{M}_{t-1}(p).\bar{I}_t(p) + [1-\overline{M}_{t-1}(p).\bar{I}_t(p)][M_{t-1}(p)+\tilde{I}_t(p)] \tag{2}$$

Where the coefficients, are the blending coefficients and the binary masks and are defined as follows:

$$\begin{aligned} &\bar{I}_t(p) = 0 \ \textit{if} \ \tilde{I}_t(p) = 0 \ (\textit{i.e., undefined}) \qquad \bar{I}_t(p) = 1 \ \textit{elsewhere}. \\ &\overline{M}_{t-1}(p) = 0 \ \textit{if} \ M_{t-1}(p) = 0 \ (\textit{i.e., undefined}) \quad \overline{M}_{t-1}(p) = 1 \ \textit{elsewhere}. \end{aligned} \tag{3}$$

1.2 Background Mosaic Computation

In dynamic mosaicing approach, after extracting the areas corresponding to the fixed background from each original image (using motion segmentation approaches), the warping parameters which permit to represent them in the mosaic reference system have to be estimated. This process is described in the following (see Fig. 1).

1.2.1 Object-Based Motion Segmentation

The original images may contain objects which should not be used to construct the mosaic image. This is obviously the case for moving objects. For instance, in the case of a panoramic view of a street, depending on the application, the cars moving on the street should not be incorporated in the mosaic image. For this reason, we use a moving object-based segmentation algorithm [7].

1.2.2 Warping Parameters Estimation

The design of a reliable warping parameters estimation algorithm is very important because even small estimation errors may generate visible artifacts mainly at the boundaries between the updated and no updated mosaic areas. In this paper, we have adopted the robust and efficient method for warping estimation proposed in [8].

1.2.3 Moving Object Insertion in Background Mosaic

In this step, in order to update our wide mosaic scene, we need a method for inserting and blending moving objects in the constructed static background mosaic (the sprite). This is important to use a basic model for moving object inserting and blending.

One of the best and most suitable models are the rectangular models which extrapolate the moving objects in the scene and warping transformations on them are very fast and easy (see Fig. 2). Finally we build a binary image that indicates the moving parts of each original image.

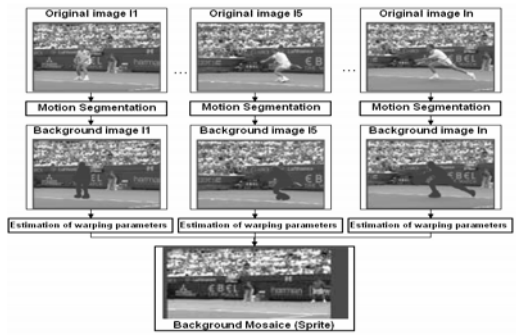


Fig. 1. General block diagram of static background mosaic calculation

2 Image Blending

Assuming that the moving objects have already been aligned, there are two main approaches for image blending in the literature.

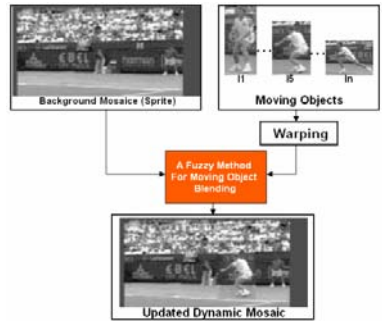


Fig. 2. Dynamic mosaic updating

The first approach are optimal seam algorithms [9], search for a curve in the overlapped region on which the differences between input images are minimal.

The second approach minimizes the seam artifacts by smoothing the transition between the image regions such as feathering or alpha blending methods [10,11]

Next section presents a novel method to blend the moving objects in static background mosaic (sprite) based an adaptive alpha blending method with respect to quality criterions (See Fig. 2). In order to compute the best possible value for alpha blending coefficient in dynamic mosaic update process, we propose a fuzzy method based on relative speed and scale of moving objects.

3 Proposed Method

In this section different parts of the proposed algorithm are explained in detail.
There are many challenges in developing a good blending algorithm. The main criterions for developing a robust and efficient blending algorithm include:

I) Minimization of temporal variations in background. The background information contained in the current compensated image can be different from the existing mosaic image. As a consequence, the mosaic images M_t and M_{t-1} can be significantly modified in the area updated. If these changes are too strong, they will degrade the temporal quality of the mosaic sequence by artificially introducing strong temporal variations and artificial boundaries at the limits areas.

II) Minimization of temporal delay. A mosaic image is not temporally homogeneous; since it is calculated using several images. As a consequence, a temporal delay can be associated to each pixel. For many applications, it will be important to have the temporal mosaic image as close as possible to the current image.

III) Being very fast and perform in real-time. In order to meet all of the mentioned criterions, here we have proposed a fuzzy approach in which alpha blending coefficient is obtained adaptively with respect to the relative speed and scale of moving objects.

3.1 Moving Object Blending

In this method, we have selected some related motion parameters based on blending criterions. From the experiments we have concluded that the speed of moving objects in a sequence and relative scales of each moving object have the most effect on blending criterions. Therefore, we have defined the Speed and Scale parameters of each moving object as a basis for our adaptive alpha coefficient calculation.

As mentioned in Section 2.2.3, applying the warping parameters on rectangular model is fairly fast and suitable. In these models, some pixels which are not belong to moving object regions and extrapolated with model boundaries have important role in blending process.

In our proposed method, to create dynamic mosaic M_t from the scene, we use a weighted combination of these pixel values and value of same pixels in static mosaic background M_{t-1} (See relation 4). The weighting coefficients (alpha) vary as a fuzzy function of related motion parameters of moving objects.

$$M_t = (1-\alpha)M_{t-1} + \alpha\tilde{I}_t \tag{4}$$

3.1.1 Relative Speed of Moving Objects

Based on our experiments, there is a direct relation among moving object relative speed and temporal variation of background information in current image. When the relative speed of a moving object in a scene increases, the temporal variation of background information increases as well.

According to this fact, we introduce a method to calculate the relative speed of moving object model in image It. In our method, we suppose that the maximum temporal movement of each moving object in image it is R, which is equal to the diameter of rectangular model, but it's obvious that every movement which is greater than R considered as fast motions and all of fast motions replaced with R value in our calculations. Then we can define the relative speed (0<Vr<1) for each moving object model as the object model movement proportion to maximum model movement R during two consequent images in the sequence. This is illustrated in Fig.4.

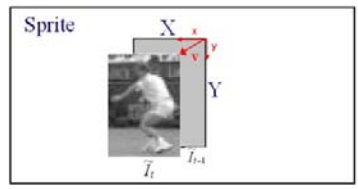


Fig. 4. Moving object relative speed calculation

Equation (5) describe how to calculate the relative speed ratio Vr for moving object models

$$R = \sqrt{X^2 + Y^2} \qquad v = \sqrt{x^2 + y^2} \qquad V_r = \frac{v}{R} \tag{5}$$

3.1.2 Relative Scale of Moving Objects

The relative scale of moving objects in the input image sequence is one of the most effective parameters in the blending process. Based on our experiments, there is an inverse relation between the amount of background information in the object model and its alpha blending value. In fact, when we have little background information in our model we give them most priority to display (bigger alpha blending value). In this method, we define the relative scale of moving object with an inverse relation with the amount of background information in our moving object rectangular model. Therefore, the relative scale parameter of a moving object represents in range [0-1] as:

$$Scale = \frac{Object\,Area}{Model\,Area} \tag{6}$$

3.2 Adaptive Alpha Blending Coefficient

In order to obtain the most appropriate value for alpha blending coefficient adaptively, we complete our fuzzy inference system with fuzzification in our outputs (alpha blending coefficients) and design an adequate rule based on our knowledge.

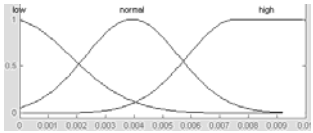


Fig. 5. Speed membership function

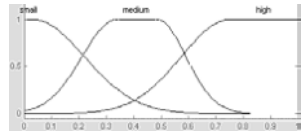


Fig. 6. Relative scale membership function

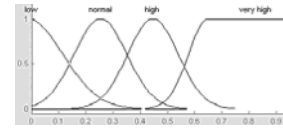


Fig. 7. Alpha coefficient membership function

One of the most important steps in our method is how to configure our rule base. Our experimental results showed that our output values (alpha blending coefficients) are more sensitive to the relative scale of moving objects. We implemented this property in rule 6, 7 in our system rule base (See Fig. 8). Also, based on our empirical results, we concluded that the best behavior for alpha blending coefficient generator is an exponential behavior (see Fig. 9).

if (*Speed* = *Low*) & (*Scale* = *Small*) ⇒ (*Alpha* = *Low*)
if (*Speed* = *Low*) & (*Scale* = *Medium*) ⇒ (*Alpha* = *Low*)
if (*Speed* = *Low*) & (*Scale* = *Big*) ⇒ (*Alpha* = *Normal*)
if (*Speed* = *Normal*) & (*Scale* = *Small*) ⇒ (*Alpha* = *Low*)
if (*Speed* = *Normal*) & (*Scale* = *Medium*) ⇒ (*Alpha* = *Normal*)
if (*Speed* = *Normal*) & (*Scale* = *Big*) ⇒ (*Alpha* = *High*)
if (*Speed* = *High*) & (*Scale* = *Small*) ⇒ (*Alpha* = *Normal*)
if (*Speed* = *High*) & (*Scale* = *Medium*) ⇒ (*Alpha* = *High*)
if (*Speed* = *High*) & (*Scale* = *Big*) ⇒ (*Alpha* = *High*)

Fig. 8. Our fuzzy rule bases

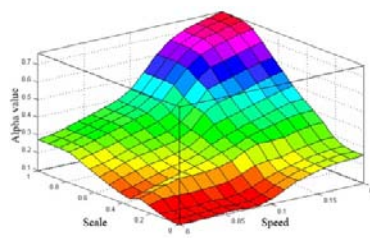


Fig. 9. Fuzzy surface of our system

4 Experimental Results

Fig. 10 shows the mosaic image obtained of the well known “Stefan” sequence with the comparison of proposed method with constant alpha blending method (Alpha=0.8). Fig. 10.d and f, show the obtained wide scene of the “Stefan” sequence which is constructed by dynamic mosaicing method. Then, moving objects are blended on them with constant alpha blending method and adaptively alpha blending coefficient which is proposed in this paper. In Figs. 10.e and f, for easier determination of differences among two results, we have zoomed on the blended moving object in the obtained dynamic mosaic images.

In Fig. 10.e, you can find some critical misalignment artifacts around the moving object (look at the chair that overlapped with the moving object). But, in the same region in Fig. 10.g, you can find that the errors of misalignments smoothed in distance between moving object and rectangular model boundaries.

Fig. 11 presents the constructed dynamic mosaic image of standard sequence “Indoor PETs 2002”. One type of artifacts that often occurs in dynamic mosaic construction is ghosting effect; these ghosting effects occasionally happen in dynamic mosaics because of some misalignments in moving object registration or motion segmentation

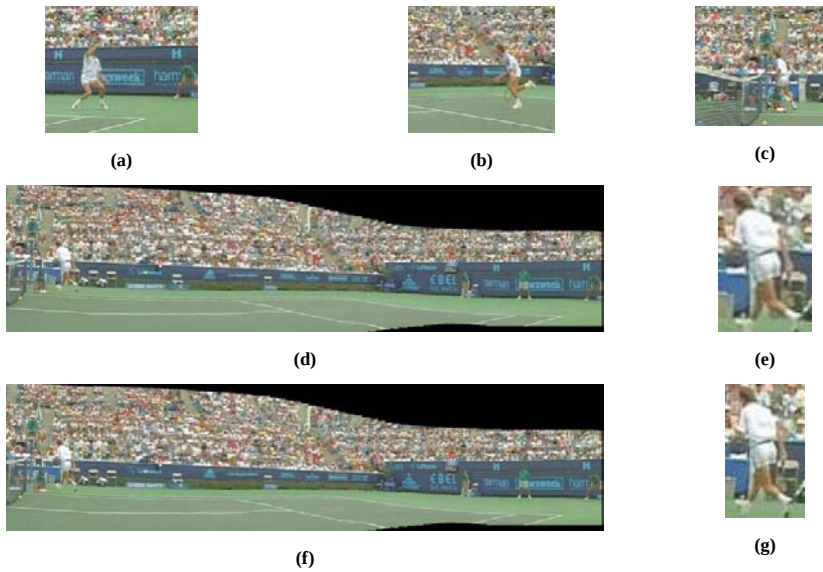


Fig. 10. (a, b, and c) Original images of input "Stefan" sequence. (d) Blended moving object on the background mosaic with constant alpha blending coefficient (Alpha = 0.8). (e) Zoomed on moving object shown in (d). (f) Blended moving object using proposed method. (g) Zoomed on moving object in shown in (f).

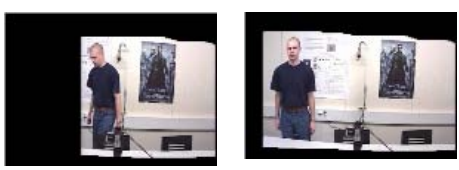


Fig. 11. Constructed scene by proposed dynamic mosaicing method for "Indoor PETs 2002" sequence.

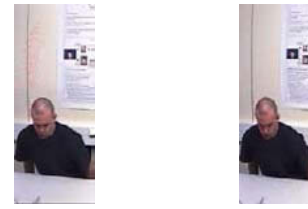


Fig. 12. (a) Ghosting effect in constant alpha value. (b) Removal of ghost effect by proposed

Table 1. Time complexity comparison (ms/f)

Video Type	Video Format	Pyra-mid Blend-ing	GIST	Constant Alpha Blend-ing	Propose Method
"Stefan"	CIF	1894	2099	28	120
"Indoor PETs"	CIF	2254	2584	29	208
"Bus"	CIF	2018	3602	21	156

algorithms. Fig. 12 shows that the ghosting effects are removed when using the proposed adaptive alpha blending coefficient values based on object speeds.

As you can see in this table, the time complexity of the "GIST"[12] and Pyramid blending methods [11] are too high for real-time applications. In average, the resulting mosaic qualities of these methods are visually better than the alpha blending methods. But due to their high computational cost they are more appropriate for off-line image blending applications.

5 Conclusion

In this paper, we proposed a novel method for moving object blending in dynamic mosaic images based on fuzzy approximation of blending coefficient value adaptively. Here we proposed a fuzzy method based on relative speed and scale parameters of moving objects to calculate alpha blending coefficient. The experimental results showed that the proposed blending method is a good candidate for real-time applications and introduced suitable trade off between computational cost and visual quality.

Acknowledgment

This work was in part supported by a grant from ITRC.

References

1. Sawhney, H.: Compact representation of video through dominant and multiple motion estimation. IEEE Trans. Pattern Anal. Machine Intell. 18, 814–830 (1997)
2. Kumar, R., Irani, M., Anandan, P., Bergen, J., Hsu, S.: Efficient representations of video sequences and their applications. Signal Process. Image Commun. 8, 327–351 (1996)
3. Bonnet, M.: Mosaic representation for video shot description. In: Proc. MPEG-7 Evaluation Ad Hoc Meeting, p. 636 (February 1999)
4. Szeliski, R.: Video mosaic for virtual environment. Comput. Graph. Applicat., 22–30 (March 1996)
5. Nicolas, H.: New Methods for Dynamic Mosaicking. IEEE Trans. Image Processing 10, 1239–1251 (2001)
6. Irani, M., Anandan, P.: Video Indexing based on Mosaic Representation. Proc. IEEE 86, 905–921 (1998)
7. Bagheri, M., Lotfi, T., Darabi, A.A., Kasaei, S.: Content-Based Video Coding for Distance Learning. In: The 7th IEEE International Symposium on Signal Processing and Information Technology, ISSPIT, Cairo, Egypt (December 2007)
8. Lorei, M., Smolis, A., Sikora, T.: Adaptive Kalman filtering for prediction and global motion parameter tracking of segments of video. In: Proc. PCS (1997)
9. Agarwala, A., Dontcheva, M., Agrawala, M., Drucker, S., Cohen, M.: Interactive digital photomontage. ACM Trans. Graph. 23(3), 294–302 (2004)
10. Uyttendaele, M., Eden, A., Szeliski, R.: Eliminating ghosting and exposure artifacts in image mosaics. In: Conf. on Computer Vision and Pattern Recognition, pp. II, 509–516 (2001)
11. Adelson, E.H., Anderson, C.H., Bergen, J.R., Burt, P.J., Ogden, J.M.: Pyramid method in image processing. RCA Engineer 29(6), 33–41 (1984)
12. Zomet, A., Levin, A., Peleg, S., Weiss, Y.: Seamless image stitching by minimizing false edges. IEEE Trans. on image Processing 15(4), 969–977 (2006)

Inferring a Bayesian Network for Content-Based Image Classification

Shahriar Shariat[1], Hamid R. Rabiee[2], and Mohammad Khansari[3]

[1] AICTC Research Center, Sharif University of Tech., Computer Engineering Department, Digital Media Research Lab
s_shariat@ce.sharif.edu
[2] AICTC Research Center, Sharif University of Tech., Computer Engineering Department, Digital Media Research Lab
rabiee@sharif.edu
[3] AICTC Research Center, Sharif University of Tech., Computer Engineering Department, Digital Media Research Lab
khansari@mehr.sharif.edu

Abstract. Bayesian networks are popular in the classification literature. The simplest kind of Bayesian network, i.e. naïve Bayesian network, has gained the interest of many researchers because of quick learning and inferring. However, when there are lots of classes to be inferred from a similar set of evidences, one may prefer to have a united network. In this paper we present a new method for merging naïve networks in order achieve a complete network and study the effect of this merging. The proposed method reduces the burden of learning a complete network. A simple measure is also introduced to assess the stability of the results after the combination of classifiers. The merging method is applied to the image classification problem. The results indicate that in addition to the reduced computation burden for learning a complete network, the total precision is increased and the precision alteration for each individual class is estimable using the measure.

Keywords: Bayesian network, naïve networks, learning, inference, image classification.

1 Introduction

Classification is a traditional problem in pattern recognition for over last fifty years [1]. On the other hand, emergence of new technologies produced a large amount of multimedia data. Within the multimedia formats, image comprises the simplest yet very important one. Image classification has gained many researches during last two decades. Classifying and clustering alleviates both browsing and retrieving burdens from image datasets.

One of the most popular classifiers is Bayesian network. Bayesian networks have some features, which make them suitable for classification [0]. These properties have encouraged lots of researchers in diverse areas to employ Bayesian networks for classification.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 211–218, 2008.

Bayesian approach for image classification has been reported in numerous papers in literature. A Bayesian framework is constructed by Vailaya et al. to semantically classify the outdoor images [0]. They estimate the class conditional density using a code book which is extracted by a vector quantizer. Naphade and Huang in [0] combined HMM, EM and Bayesian network to first estimate the probability density for each category (Multiject) and then connect them together to enhance the classification in addition to incorporate the inter-conceptual relations between the classes. However, they have designed their framework for video but the idea could be employed for image too.

In many real-world cases, however, the experts prefer to use naïve Bayesian networks (NBN) and especially Gaussian naïve Bayesian networks (GNBN) (Fig. 1). In [0] naïve Bayesian networks are the basis of structural learning of a Bayesian network to classify genetic abnormalities. A variation of naïve Bayesian networks called semi-naïve Bayesian network is studied and another approach is introduced in [0] based on finite mixture models to combine the attributes. The authors show that this type of combination and bounding model improves the accuracy of classification. In [0] maximal covariance criterion is utilized to distinguish the necessary hidden nodes and insert them between the features of some naïve Bayesian network to classify textures of aerial images.

The usual case happens when there are lots of classes that cannot be combined in a single node and the inference is based on a similar set of evidence nodes. In this case, it would desirable to have a network consisting of all classes. Inferring from that network classifies all samples in a single inference. Thus, merging these naïve networks reduces the computational burden for the inference. Nevertheless, the learning process of a complete network needs a noticeable computation time. Approximating the parameters for a complete network seems to be unneeded, since the naïve network parameters are already learned and another learning process is not favorable. Therefore, a solution to attain the accurate parameters without performing an additional learning process will be very beneficial.

In this paper we have proposed a new framework for combining naïve Bayesian classifiers in order to achieve a complete network with direct parameter estimation from individual classifiers without accomplishing any extra learning for the combined network. We have also studied the effect of combining naïve networks and proved that the overall precision of the classifier can be improved especially when a sample belongs to one and only one class. We have also introduced a measure to estimate the stability of the classification precision after the merging. The classifier merging algorithm has been applied to the image classification problem to experiment the algorithm in practice.

The rest of paper is organized as follows. Section two presents the network architecture. The third section gives details on learning naïve network and general Bayesian networks. In the forth section the inference of the networks and the combination effect are investigated. Section five utilizes the algorithm for image classification and shows the experimental results respectively. Finally, section six concludes the paper.

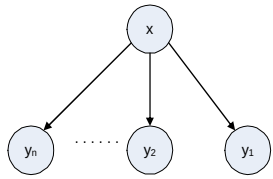


Fig. 1. Naïve Bayesian network

2 Learning

Here in the problem of image classification we have continuous evidence nodes (i.e. y_1, y_2, …, y_n in fig. 1) and multinomial class node (i.e. *X* in fig. 1). We convert the multinomial node to continuous. To simplify the learning and inference we take *X* as binary rather than multinomial and thus if *X* represents *m* classes, it is broken up to *m*-1 two-state-class nodes. Throughout the learning process the covariance matrix will be computed algorithm from [0]. For fig. 1, the covariance matrix will be:

$$\psi = \begin{pmatrix} \sigma_x & b_{y_1x}\sigma_x & b_{y_2x}\sigma_x & .... & b_{y_nx}\sigma_x \\ b_{y_1x}\sigma_x & \sigma_{y_1}+b_{y_1x}^2\sigma_x & b_{y_2x}b_{y_1x}\sigma_x & .... & b_{y_nx}b_{y_1x}\sigma_x \\ : & & & & : \\ b_{y_nx}\sigma_x & & .... & & \sigma_{y_n}+b_{y_nx}^2\sigma_x \end{pmatrix}$$

where $b_{y_i x}$ is the value of the link which connects node *X* and y_i. The mean values are obviously, the average of each node's samples.

The learning process of a naïve Bayesian network is not that hard. Looking at the covariance matrix, one can derive a very simple approach to learn a naïve network. However, merging naïve networks result in a much more complex precision matrix which is burdensome to inverse and complicated to solve. The following theorem provides a much simpler solution which, by the way, keeps the accuracy of learned parameters.

Theorem 1- Under horizontal conjunction of two or more naïve Bayesian networks provided that there is no connection between class nodes and class nodes are not descendent of any other node, each class node's variance is multiplied by

$$m = \frac{v + M - N_t}{v + M - N_n} \tag{1}$$

and thus

$$\sigma_x^t = m\sigma_x^n \tag{2}$$

where σ_x^t is the new variance, σ_x^n is X's variance in the naïve network, *M* denotes the number of training samples, N_t is the size of whole graph and N_n is the size of the

naïve network and *v* is the size of hypothetical sample upon which we base our prior belief. Moreover, the variance of evidence nodes are changed regarding the following

$$\sigma_x^t = m(\sigma_x^n - \sum_{y \in Pa_x} b_{xy}^2 \sigma_y^n) \tag{3}$$

where m is calculated from Eq. (1).

Proof:

from [0] we know that

$$(T^*)^{-1} = \frac{(v^*+1)}{v^*(\alpha^* - n + 1)} \beta^* \tag{4}$$

Calculating the parameters of the network is easy using precision matrix algorithm and solving the equation $\psi = (T^*)^{-1}$.

While connecting some naïve Bayesian networks, the only altered parameter is *n*, i.e. the number of nodes. The parameters which are affected by prior belief are not dependent on the size of network and remain unchanged. Doing some algebra we conclude that each element of $(T^*)^{-1}$ is multiplied by Eq. (1). By the way, looking at ψ, it is obvious that connector values are not changed. Again, having a look at ψ apprises us that the root nodes are changed using Eq. (1) and the other nodes by Eq. (3) .

Inference

For a naïve Bayesian network, if data is indicated by *a* then Using d-separation property and some properties of normal distributions we can write

$$P(x \mid a) = N(x; \frac{\sum_{i=1}^{n} \frac{\mu_i}{\sigma_i}}{\sum_{i=1}^{n} \frac{1}{\sigma_i}}, \frac{1}{\sum_{i=1}^{n} \frac{1}{\sigma_i}}) N(x, \mu_x, \sigma_x) \tag{5}$$

The above equation works for naïve Bayesian networks but for general networks, the non-descendent nodes affect each other and Eq. (5) does not make sense. In the following we have mentioned the two methods of classifier reconstruction.

For the first method, one can use each classifier separately and save the result of each class for '1' and '0' (the two states of the class node). This leads to the best result that a single graph can achieve and provides an ordinary classification. For further references we call this method INF1.

For the second method, one may consider reconstructing the whole graph. Actually, this method, which we refer to it as INF2, does not perform worse than INF1 and in some cases performs much better. INF2 is especially suitable for the case in which except for one class node all the rest are zero. In this case INF2 usually performs better than INF1, due to the bias of other nodes which tend to be zero. In fact they help the $\mu_{X|a}$ to shift to its target value, i.e. 1. In a singly connected network which has

only two levels and one of them consists of the evidence nodes, Pearl's [0] belief propagation algorithm can show why INF1 and INF2 perform similarly.

Utilizing conditional probability and Bay's theory it is trivial to prove that

$$P(x \mid a) = P(x \mid d_x, n_x) = \beta P(d_x \mid x) P(x \mid n_x) \tag{6}$$

and if $\lambda(x) = P(d_x \mid x)$ and $\pi(x) = P(x \mid n_x)$ where both $\lambda(x)$ and $\pi(x)$ are normal random variables. According to [0].Their parameters are given by

$$\mu_x^{\lambda} = \frac{\sum_{i=1}^{n} \frac{\mu_i}{\sigma_i}}{\sum_{i=1}^{n} \frac{1}{\sigma_i}} \quad and \quad \sigma_x^{\lambda} = \frac{1}{\sum_{i=1}^{n} \frac{1}{\sigma_i}} \tag{7}$$

$$\mu_x^{\pi} = \mu_x \quad and \quad \sigma_x^{\pi} = \sigma_x \tag{8}$$

The above result for naïve network comply with Eq (5).

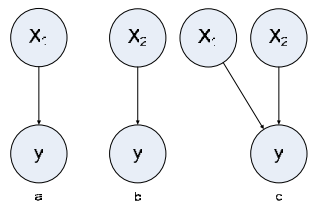


Fig. 2. (a) and (b) Simplest connections, (c) combination of a and b

Now to observe the effect of cascading two networks like Fig. 1, we start with the simplest possible case in fig. 2. In Fig 2(a) and Fig 2(b) we are in the same situation as in Fig 1. For Fig 3(a) we have

$$\mu_{x_1}^{\lambda} = \frac{y}{b_{x_1 y}} \quad and \quad \sigma_{x_1}^{\lambda} = \frac{\sigma_y}{b_{x_1 y}^2} \tag{9}$$

and similar equations are true for Fig 2(b). From a belief propagation algorithm one can calculate the above values as

$$\mu_{x_1}^{\lambda} = \frac{y - b_{x_2 y} \mu_{x_2}}{b_{x_1 y}} \; and \quad \sigma_{x_1}^{\lambda} = \frac{\sigma_y}{b_{x_1 y}^2} \tag{10}$$

In the case we stated in this section where class nodes are all zero but one, the mean value of class nodes are not large and if the proportion of the connector values are not large either, then the value of $\mu_{x_1}^{\lambda}$ in Fig 2(c) is not far different form that of shown in Fig 2(a). In larger networks the relations are more complicated and actually the effect of other class nodes is weaker. We also need a kind of measure that assesses

the correctness of this combination. The problem is the difference between $\mu_{x_1}^{\lambda}$ in fig. 3(a) and (b). The following formula provides estimation for the expected variation in class node mean value using Eq. (10). In the L_measure equation (Eq. (11)) N_n is the number of evidence nodes connected to each class node (n) . in Eq. (11) the term, $p \in parent(v)$ indicates the nodes which are v's parents in the merged network. The returned value is a measure and using a threshold, one can accept or reject INF2. Usually L<1 is a good threshold for L.

$$L_n = \frac{\sum_{v \in child(n)} \left| \sum_{p \in parent(v), v \neq t} \frac{b_{pv}\mu_p}{b_{vn}} \right|}{N_n} \tag{11}$$

3 Experimental Results

We used 4 concepts of Corel 5.0 image database. The images were fed to the segmenting algorithm [0]. After all images are segmented, the feature extractor engine runs and 16 features which are all based on texture and color moments are extracted from the image segments.

The images collected from Corel 5.0 dataset is divided randomly into three parts; two parts as the training data and one part for test. We have studied the classification precision for the finest classes.

Table 1. L Measure

Class name	**L *measure***
Sky	.1573
Building	.6205
Elephant	.7691
Grass	4.5282

Table 2. INF 1

Class name	***Precession***
Sky	96.34%
Building	88.38%
Elephant	72.93%
Grass	77.90%
Total	67.44%

L measures are presented in Table 3. One can see that for the first three classes, L is less than one. Table 4 and 5 show that for lower L measures the precision for INF1 and INF2 are not different that much.

Table 4 shows the implementation results for INF1 and table 5 for INF2. The precision calculated for each class in independent from other classes. Although, the precision for individual classes are not that different (except for grass, as L measure suggests) the interesting point is the total precision which is dramatically increased by INF2.

Table 3. INF2

Class name	***Precession***
Sky	96.34%
Building	90.68%
Elephant	75.72%
Grass	70.41%
Total	79.40%

4 Conclusion

In this paper we have studied the effect of merging Bayesian naïve classifiers in order to alleviate the inferring burden and increasing the total classification precision. The proposed merging method omits the additional learning phase for the complete network and results in a less computation.

The introduced stability measure for the classifiers precision (L measure) has been applied to an image classification problem. This measure can be calculated for each classifier. In other words, the researchers can use this measure to decide in which classifiers the precision will not change considerably after merging.

The experimental results showed that total classification precision for a 4 class image classification problem after merging the naïve classifiers has been increased by 10 to 12 percent. Moreover, for the classes with L measure less than 1 the merging does not affect the individual classifier precision too much.

Currently, our classifier merger does not support the causal connection between class nodes and also the hierarchical networks. These two issues can be considered as the future works. Moreover, the merging quality measure (L measure) does not directly and quantitatively express the precision alteration and it is a qualitative measure. Finding a more accurate and quantitative measure could also be a good research topic for future.

Acknowledgement

This research has been funded by the Advanced Information and Communication Technology research Center (AICTC) of Sharif University of Technology.

References

1. Jain, A.K., Duin, R.P.W., Mao, J.: Statistical pattern recognition: A review. IEEE Trans on pattern recognition and machine intelligence 22(1), 4–37 (2000)
2. Heckerman, D.: A tutorial on learning with Bayesian networks. Technical report, Microsoft Research (1996)

3. Vailaya, A., Figueiredo, M., Jain, A., Zhang, H.J.: Image Classification for Content-Based Indexing. IEEE Transactions on Image Processing 10(1), 117–130 (2001)
4. Naphade, M.R., Huang, T.S.: A probabilistic framework for semantic video indexing, filtering and retrieval. IEEE Trans. on multimedia 3(1), 141–151 (2001)
5. Lerner, B., Malka, R.: Learning Bayesian Networks for Cytogenetic Image Classification. In: Proc. 18th ACM conf. on pattern recognition, vol. 2, pp. 772–775 (2006)
6. Huang, K., King, I., Lyu, M.R.: Finite mixture model of bounded semi-naive Bayesian network classifier. In: Kaynak, O., Alpaydın, E., Oja, E., Xu, L. (eds.) ICANN 2003 and ICONIP 2003. LNCS, vol. 2714, pp. 115–122. Springer, Heidelberg (2003)
7. Yu, X., Zheng, Z., Wu, J., Zhang, X., Wu, F.: Texture classification of aerial image based on Bayesian networks with hidden nodes. In: Advances in computation and intelligence. Springer, Heidelberg (2007)
8. Chickering, D.M., Heckerman, D.: Efficient approximation of the marginal likelihood of Bayesian networks with hidden variables. Technical report, Microsoft Research (1997)
9. Shachter, R.D., Keneley, D.: Gaussian influence diagrams. Management science 35 (1989)
10. Degroot, M.H.: Optimal Statistical Decisions. McGraw-Hill, New York (1970)
11. Pearl, J.: Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann, San Francisco (1988)
12. Bouman, C.A., Shapiro, M.: A multiscale random field model for Bayesian image segmentation. IEEE Trans. on image processing 3(2), 162–174 (1994)

High Performance Mathematical Quarter-Pixel Motion Estimation with Novel Rate Distortion Metric for H.264/AVC

Somayeh Sardashti, Hamid Reza Ghasemi, Mehdi Semsarzadeh, and Mahmoud Reza Hashemi

Multimedia Processing Laboratory, School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran
ssardashti@cad.ece.ut.ac.ir, hrghasemi@cad.ece.ut.ac.ir, m.semsar@ece.ut.ac.ir, rhashemi@ut.ac.ir

Abstract. Fractional Motion Estimation (FME), which refines integer motion vectors found during Integer Motion Estimation, is one of the most time and computation consuming parts of the H.264 encoder. Conventional FME performs the time and resource consuming sub-pixel interpolation and subsequent secondary search, for each integer motion vector. In this paper using a novel rate distortion cost metric for sub-pixel selection, we introduce an improved mathematical method which estimates SAD values at quarter-pixel positions with far less computation compared to the conventional approach. The proposed scheme reduces both computation time and memory access requirements without any significant quality degradation. Simulation results indicate that the proposed method improves both PSNR and bit rate by 1 dB and %18 respectively compared to the basic mathematical method for fast motion samples. In the case of slow motion samples, it increases PSNR up to 3 dB.

Keywords: H.264, Fractional Motion Estimation, Integer Motion Estimation.

1 Introduction

Recent advances in digital wireless networks and improvements in multimedia compression algorithms have made the transmission of video streams over mobile networks a reality. The wide range of applications and the higher compression efficiency of H.264/AVC make it the method of choice for mobile video applications [1]. The H.264/AVC is the latest international video coding standard jointly developed by the ITU-T Video Coding Experts Group and the ISO/IEC Moving Picture Experts Group in 2003. It achieves a higher coding efficiency compared to the previous standards such as MPEG-1,-2,-4 and H.263. This improved performance has been realized partly due to the new context based entropy coder, but also because of several prediction enhancements, namely: variable block-size motion estimation, intra prediction, multiple reference frames and sub-pixel motion estimation.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 219–226, 2008.

Unlike the previous video coding standards which use constant macro-block sizes, H.264/AVC allows variable macro-block sizes for motion estimation (ME) and compensation (MC) which improves the coding efficiency by up to 15%. The possible macro-block sizes in H.264/AVC are 16x16, 16x8, 8x16, 8x8, 8x4, 4x8, and 4x4. Furthermore, 20% bit rate saving is achieved by supporting up to 16 reference frames compared to two in the previous standards. It also allows for up to eighth pixel fractional motion estimation (FME), compared to quarter pixel FME in MPEG-4, which improves the bit rate from 5% to 20% [2].

However, the improvements are obtained at the expense of higher complexity and computational load especially from the H.264/AVC motion estimation block which takes up to 80% of the total encoding time [3]. Considering the importance of power consumption in mobile applications this block may become one of the major system bottlenecks.

Conventional FME method requires time consuming half-pixel and quarter-pixel interpolation. The interpolation unit needs large amount of memory access and is computationally intensive, which is a major challenge for real time applications. There are half-pixel accuracy searching methods [4] to decrease complexity caused by interpolation. But these methods commonly lead to large estimation errors due to simplified error models. In [5] and [6] another method is proposed which is based on mathematical models of the mean-square MC prediction errors. This method gains a performance close to the conventional method for sequences containing average to fast content, while having a far less computational complexity.

Although the proposed mathematical method in [5] and [6] performs close to reference software for some video samples, they impair the quality in slow videos. To address this problem, we have proposed a novel rate distortion cost metric for the mathematical method which improves the results for all video content from slow to fast.

The rest of this paper is organized as follows. The basic mathematical FME method and its related problems are discussed in section 2. The proposed improved fractional motion estimation method is introduced in section 3. Experimental results are presented in section 4. Finally, section 5 concludes the paper.

2 Related Work

The conventional FME method consists of two main complex tasks: interpolation of reference frames and searching for the best interpolated point around the pixel pointed by the Motion Vector (MV) found during Integer Motion Estimation (IME). In order to avoid these complex tasks, reference [5] introduces a mathematical method which estimates Sum of Absolute Differences (SADs) at half-pixel precision according to neighboring integer-pixel precision SADs. This method avoids half-pixel interpolation and reduces computation time. Reference [6] extends this method to quarter-pixel precision. Experimental results show that the performance of these methods is close to the conventional approach, while having lower computational complexity [5], [6].

Close scrutiny of this basic mathematical method and applying it to video samples with slow content, however, reveals that it results in much lower quality in comparison with the conventional FME approach. In slow motion videos, where the Integer MVs (IMVs) are smaller, selecting a refined MV at half- or quarter-pixel level different from the one found by the reference software results in more quality degradation comparing to the same situation with fast videos where the IMVs are larger.

This degradation is caused by using an improper comparison metric to select among different search points at half and quarter-pixel levels. To solve this problem, in this paper, a new rate distortion cost metric has been introduced that improves the efficiency of the mathematical method. Simulation results show that the proposed technique has up to 1 dB PSNR improvement over existing mathematical models in fast motion samples. More importantly, while the method of references [5] and [6] destroys quality in slow motion videos, the introduced scheme gains up to 3 dB PSNR improvement.

```
Loop1 (Reference Frame)
  Loop2 (41 MV of the sub-blocks)
    Loop3 (Half/Quarter)
      Loop4 (9 search points around the selected Integer/Half pixel) {
            sub-pixel interpolation of the sub-block
            Residue Generation
      }
```

Fig. 1. Four loops of FME procedure

3 Proposed FME Algorithm

In order to refine integer motion vectors, which are calculated in IME part, at half or quarter precision, FME goes through four iterations. Fig. 1 shows these iterations which are implemented in the reference software in order to compute fractional motion vectors (FMVs) [7]. The first two loops are sub-blocks from different block types and reference frames that are selected to form a predicted Macro Block (MB). The third loop sequentially processes each sub-block in half and then in quarter precision. The last loop includes interpolation and residue generation for nine candidates around the best integer or half pixel and selecting the best candidate.

Therefore, for each MB, it is required to interpolate its related search area in all reference frames which is a time and memory consuming procedure. References [5] and [6] have proposed to estimate SADs at half-pixel precision based on the neighboring integer-pixel SADs, and therefore they avoid sub-pixel interpolation, and reduce computation time and complexity. In this paper, we apply this method to both half-pixel and quarter-pixel precisions with an improved rate distortion cost metric. The details are presented in the following subsections.

3.1 Half-Pixel Precision FME

In Fig. 2, circles denote integer pixels and squares depict half pixels around the origin which is the integer pixel pointed by the integer motion vector selected during IME.

The error plane defined by these nine integer pixels can be modeled by the following Formula [5].

$$f(x,y)=\sum_{i=0}^{2}\sum_{j=0}^{2}C_{10-(i+1)*(j+1)}x^{i}y^{j} \tag{1}$$

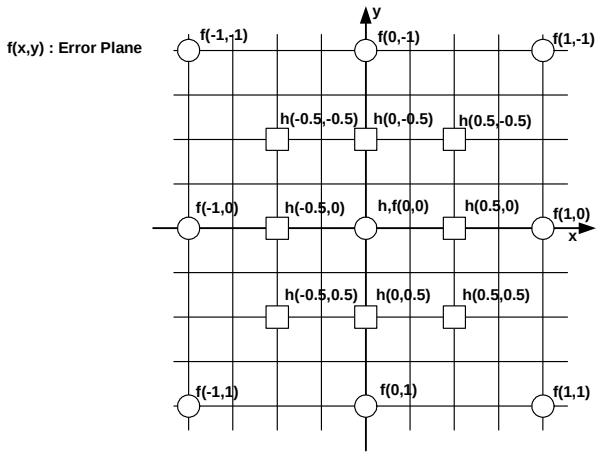


Fig. 2. Integer and half pixels

The coefficients here can be determined by substituting nine integer-pixel precision SADs around the origin sequentially from the upper left position, f (-1,-1), to the lower right position, f (1,1) of Fig. 2. The matrix derived from replacing integer-pixel SADs and their coordinates in (1), is shown in Matrix (2) [5].

$$\begin{bmatrix} f1 \\ f2 \\ f3 \\ f4 \\ f5 \\ f6 \\ f7 \\ f8 \\ f9 \end{bmatrix} = \begin{bmatrix} 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} c1 \\ c2 \\ c3 \\ c4 \\ c5 \\ c6 \\ c7 \\ c8 \\ c9 \end{bmatrix} \tag{2}$$

Obviously, the coefficients can be calculated by the inverse matrix of (2). SADs at half-pixel points can be obtained by replacing their coordinates in (1) resulting in (3) [5], [6].

$$\begin{bmatrix} h1 \\ h2 \\ h3 \\ h4 \\ h5 \\ h6 \\ h7 \\ h8 \\ h9 \end{bmatrix} = \begin{bmatrix} 1/16 & -1/8 & -1/8 & ¼ & ¼ & -½ & ¼ & -½ & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & ¼ & -½ & 1 \\ 1/16 & -1/8 & 1/8 & -¼ & ¼ & ½ & ¼ & -½ & 1 \\ 0 & 0 & 0 & 0 & ¼ & -½ & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & ¼ & ½ & 0 & 0 & 1 \\ 1/16 & -1/8 & -1/8 & -¼ & ¼ & -½ & ¼ & ½ & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & ¼ & ½ & 1 \\ 1/16 & -1/8 & 1/8 & ¼ & ¼ & ½ & ¼ & ½ & 1 \end{bmatrix} \begin{bmatrix} c1 \\ c2 \\ c3 \\ c4 \\ c5 \\ c6 \\ c7 \\ c8 \\ c9 \end{bmatrix} \tag{3}$$

3.2 Sub-Pixel Selection Methodology

After finding SAD estimates of half-pixel candidates, the best half-pixel point is selected. In order to select the best point, there are two extreme methods. First, the cost for search points can be calculated as the method used by reference software for mode decision when Rate-Distortion Optimized (RDO) is on. It considers the distortion and

bit rate by carrying out the entire encoding loop. Using this method, the Rate Distortion cost (RDCost) is calculated for 9 half or quarter candidate pixels, and the one with the lowest cost is selected. This procedure is computationally expensive while it results in high quality and compression rate. On the other hand, there is a low complexity method which decides on SAD estimates of candidate pixels. This simple cost function is used in both references [5], and [6].

$$Cost = SAD + MVCost \tag{4}$$

In order to improve sub-pixel selection method, a proposed cost function is used which achieves a performance between the two RDO-On and SAD extreme methods. With the proposed rate distortion metric, for each candidate half or quarter pixel, cost is defined as the cost of its FMV in addition to its SAD estimate, as shown in (4). The MV cost is multiplication of Lambda factor (a function of Quantization Parameter) and the bit usage of motion vector difference (MVD) which is the difference between the current FMV and the predicted MV calculated from MVs of neighboring MBs, shown in (5).

$$MVCost = \lambda * Bit\_Usage\_Of\_MVD \tag{5}$$

As the entropy coder (CAVLC or CABAC) codes and sends the MVD and Residual values, the proposed cost function uses these factors and models them as MVCost and SAD respectively. By using this cost function the amount of MV rate will be considered as well. Therefore, this metric presents a more accurate estimate of total bit rate. This new cost metric results in much better quality than the one used in [5], and [6] and is less complex in comparison with RDO-On metric.

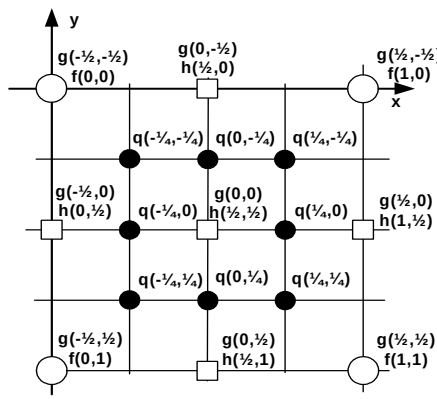


Fig. 3. Selected half pixel and its neighboring quarter pixels

Therefore, to refine an integer motion vector at half-pixel, the best half-pixel point is selected by calculating cost using (4) and (5), for all half-pixel candidates and finding the one with the minimum cost. Then, the best MV is updated to point to this best half-pixel position.

3.3 Quarter-Pixel Precision FME

In order to refine the motion vector at quarter-pixel level, we reset the origin to the best half pixel found in the previous step, and find the quarter-pixel precision by substituting the coordinates of candidate quarter pixels using (1). Four of these points are integer pixels, which their SAD values are ready from IME phase, three points are half pixels which their SAD values are calculated in the previous step, and the remaining two points are half-pixel points that are required to be calculated now. These two points can be calculated easily by replacing their coordinates in equation (1). To find quarter-pixel points around the best half-pixel point, the same process as half-pixel level should be performed. Fig. 3 shows this process when h9 (h(1/2,1/2)) is selected in the previous step as the best half-pixel position.

Finally, the best quarter-pixel point is selected by calculating cost using (4) and (5), for all quarter-pixel candidates and finding the one with the minimum cost. Then, the motion vector is updated to point to this best quarter-pixel position.

4 Experimental Results

To evaluate the improvements gained by the proposed method, three sequences 'garden', 'football', and 'tennis' are used at first. Each sequence consists of 100 frames in SIF (352x240) format. PSNR and bit rate measures are shown for the sequences with different methods and different conditions. The gained results are shown in comparison with original FME model of reference software JM 12.2 [8] (Intra Period = 15; reference frames = 2; search range = +/- 16) and with the proposed methods in [5], and [6]. For better evaluation, results have been presented for both cases where Rate Control and Rate Distortion Optimization (RDO) are on and off.

Table 1. Video Quality Comparison (bit rate)

Rate Control Off				
FME	JM 12.2	Proposed	References [5,6]	Comment
garden	2839.04	3059.53	3064.07	RDO On
	2556.01	2883.39	3174.74	RDO Off
football	2264.83	2398.97	2420.19	RDO On
	2066.14	2286.1	2735.81	RDO Off
tennis	1047.31	1128.13	1131.86	RDO On
	942.88	1091.11	1374.59	RDO Off
Average	2050.39	2195.54	2205.37	RDO On
	1855.01	2086.87	2428.38	RDO Off

Table 1 shows bit rate values of all sequences with active and inactive RDO when Rate Control is off with constant Qp = 28. When Rate Control is off, the proposed method approximately achieves the same PSNR as the method of references [5], and [6] for both states of RDO on and off. Although PSNR values remain approximately constant when Rate Control is off, the bit rate values are different for various used methods. As shown in Table 1, the improved method gets about %18 better bit rate in comparison with the methods in [5], and [6].

Table 2. Video Quality Comparison (PSNR)

Rate Control On				
FME	JM 12.2	Proposed	References [5,6]	Comment
garden	27.66	27	27	RDO On
	27.36	26.39	25.72	RDO Off
football	29.82	29.43	29.35	RDO On
	29.5	28.61	27.3	RDO Off
tennis	35	34.6	34.59	RDO On
	34.76	33.87	32.83	RDO Off
Average	30.83	30.34	30.31	RDO On
	30.54	29.62	28.62	RDO Off

In addition to studying results with inactive Rate Control, the results are considered when Rate Control is on. Table 2 shows the PSNR values of all sequences with active and inactive RDO when Rate Control is on with bit rate is set to 1 Mbps. With active RDO, the proposed method approximately gains the same PSNR as method of references [5], and [6] and about 0.05 dB less PSNR than JM 12.2. In the case of RDO off, the new method outperforms the basic mathematical method by increasing PSNR by 1 dB, while being 0.9 dB less than the JM 12.2 reference software. In this case, the proposed method approximately gains the same bit rate as method of references [5], and [6] for both states of RDO on and off.

Table 3. Video Quality Comparison (PSNR)

Rate Control On				
FME	JM 12.2	Proposed	References [5,6]	Comment
City	32.29	32.16	31.91	RDO On
	32.06	31.49	28.17	RDO Off
Crew	34.96	34.68	34.61	RDO On
	34.53	34.1	31.32	RDO Off
Harbour	28.99	28.81	28.62	RDO On
	28.77	28.38	26.89	RDO Off
Average	32.08	31.88	31.71	RDO On
	31.79	31.32	28.79	RDO Off

Three used samples of ‘garden’, ‘football’, and ‘tennis’ are considered as fast motion samples. Here, both the improved method and the basic method are applied to three other samples, ‘city’, ‘crew’, and ‘harbour’, which have less motion. Each sequence consists of 100 frames in 4CIF (704x576) format. In order to study the quality degradation, the Rate Control is considered on. Table 3 shows the results for the PSNR values of these three sequences when the bit rates are about the same (around 1 Mbps).

As shown in Table 3, when RDO is off, the basic method proposed in references [5], and [6] degrades the quality by around 3 dB. Thus, this method destroys all the quality considered to be gained from FME in these samples with little motion. In contrast, the improved method results in a much better quality, just 0.4 dB less than

JM 12.2. When IMVs are small, the effects of deviation of estimated FMVs from the actual FMVs on total quality is much noticeable. Therefore, the proposed method results in higher quality because of selecting FMVs more accurately. When RDO is on, both the developed model and the basic mathematical method gain results near the JM 12.2, because of the complex mode decision method used in the H.264 encoder.

5 Conclusions

In this paper, an improved mathematical FME is presented for H.264/AVC. In this method, instead of calculating accurate SADs at quarter-pixel, SADs are estimated with an improved rate distortion cost metric. The effect of the proposed rate distortion metric is tested in different situations (Rate Control On/Off, RDO On/Off) with different video samples. The results show that the proposed method outperforms the basic mathematical technique both in bit rate and PSNR. For fast motion samples, the proposed method improves PSNR by 1 dB and bit rate by %18 in comparison with basic mathematical method. In addition, for video sequences with slow content, the proposed scheme improves PSNR by up to 3 dB compared to the basic method.

References

1. Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 ISO/IEC 14496-10 AVC), Geneva, Switzerland (2003)
2. Shen, Y., Zhang, D., Huang, C., Li, J.: Fast Mode Selection Based on Texture Analysis and Local Motion Activity in H.264/JVT. In: 2004 International Conference on Communications Circuits and Systems, vol. 1, pp. 539–542 (2004)
3. Chen, Z., Zhou, P., He, Y.: Fast Integer Pel and Fractional Pel Motion Estimation for JVT. In: 6th Joint Video Team (JVT) Meeting, Japan, pp. 5–13 (2002)
4. Li, X., Gonzales, C.: A Locally Quadratic Model of the Motion Estimation Error Criterion Function and its Application to Subpixel Interpolations. IEEE Transaction on Circuits and Systems for Video Technology 6, 118–122 (1996)
5. Suh, J.W., Jeong, J.: Fast Sub-pixel Motion Estimation Techniques Having Lower Computation Complexity. IEEE Transaction on Consumer and Electronic 50, 968–973 (2004)
6. Kao, C.Y., Kuo, H.C., Lin, Y.L.: High Performance Fractional Motion Estimation and Mode Decision for H.264/AVC. In: International Conference on Multimedia & Expo (2006)
7. Chen, T.C., Huang, Y.W., Chen, L.G.: Fully Utilized and Reusable Architecture for Fractional Motion Estimation of H.264/AVC. In: Proc. IEEE International Conference Acoustics, Speech, and Signal Processing, pp. 9–12 (2004)
8. Joint Video Team Reference Software JM 12.2, ITU-T, `http://bs.hhi.de/suehring/tml/download`

A Versatile Reconfigurable Bit-Serial Multiplier Architecture in Finite Fields GF(2m)

Morteza Nikooghadam, Ehsan Malekian, and Ali Zakerolhosseini

Department of Electrical and Computer Engineering,
Shahid Beheshti University, Tehran, Iran
{m_nikooghadam,e_malekian,a-zaker}@sbu.ac.ir

Abstract. This paper presents a design for an efficient architecture of a reconfigurable bit-serial polynomial basis multiplier for Galois field GF(2m). The multiplier operates on the Most Significant Bit (MSB)-first for finite field multiplication. The design is flexible enough to configure different value of irreducible polynomial degree m for multiplication. Since the multiplier is doing a bit-serial processing with the gated clock technique, thus the design would be suitable for low power devices. Another advantage of the proposed architecture is the improvement of its maximum clock frequency and the high order of flexibility which allows an easy configuration for different field sizes.

Keywords: Galois field, Bit-serial multiplier, Irreducible polynomial, Linear Feedback Shift Register, Elliptic Curve Cryptography.

1 Introduction

Finite field arithmetic operations have much application in cryptography [3] and coding theory [14]. Calculations on elliptic curves involve doubling and addition of curve points. These operations are calculated by operations in the finite field GF(2m). Developing efficient arithmetic operators in GF(2m) is a real issue for Elliptic Curve Cryptosystems (ECC) [12] where the irreducible polynomial degree m of the extension must be large ($160 \leq m \leq 500$) [13]. The overall performance of an elliptic curve processor is mainly determined by the speed to multiply field elements. Hence, the multiplier will be the most important building block of such a processor.

The flexibility of our architecture forms an important advantage for cryptographic applications. The architecture introduced here is scalable to every desired bit-length and allows any choice of binary field or irreducible polynomial. We require no special properties for the irreducible polynomial defining the finite field. These properties make the proposed multiplier as a suitable choice for all applications of ECC.

We consider a $GF(2^m)$ multiplier as versatile if it works over a wide range of finite fields, i.e. a multiplier originally dimensioned for a data path precision of M bits (The value of M determines the maximum size that the multiplier can support) is also usable for fields of smaller order. The choice of the field order affects the number of points in the elliptic curve group and the difficulty of the corresponding discrete logarithm problem. A versatile multiplier for $GF(2^m)$ allows to select the degree m of the field according to the desired security requirements. For example, a multiplier

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 227–234, 2008.

designed for M = 256 bits can also be used for multiplications in $GF(2^{233})$, $GF(2^{193})$, or $GF(2^{163})$.

In this paper we introduce a bit-serial multiplier architecture that is versatile and has no restrictions on the form of the irreducible polynomial. A multiplication in $GF(2^m)$ is performed serially in a bit-by-bit fashion, starting at the most significant bit (MSB) of the polynomial multiplier. The reduction modulo the irreducible polynomial $p(x)$ is interleaved with the accumulation of partial products, which means that in each step an intermediate result of degree m has to be reduced to a polynomial of degree (at most) $m-1$. Our main contribution is a new method for degree reduction that works with any irreducible polynomial $p(x)$ of degree $m \leq M$.

The remainder of this paper is structured as follows: Section 2 provides some background information on multiplier architectures for $GF(2^m)$ and discusses related works. In Section 3, a brief description of the bit-serial architecture for multiplication in $GF(2^m)$ is given. In Section 4, the reconfigurable architecture of the proposed multiplier is presented. Test and results are presented in section 5. Section 6 concludes the paper.

2 Background and Related Work

When using a polynomial basis representation, any element of GF(2m) can be expressed as a binary polynomial of degree at most m−1, and the multiplication of field elements is performed modulo the irreducible polynomial p(x).

Several architectures have already been developed to construct the low complexity bit-serial and bit-parallel multiplications.

Bit-parallel multipliers perform a multiplication in $GF(2^m)$ in constant time (independent of m), but with an area-complexity of $O(m^2)$. Due to the large field orders used in elliptic curve cryptography, the usage of a bit-parallel multiplier in smart cards is unrealistic because of the immense hardware requirements. On the other hand, super-serial multipliers are very small, but their performance may be too poor for large field orders. Bit-serial architectures offer a fair performance/area trade-off, but they are not performance-scalable [18].

In 1989, Itoh and Tsujii designed two low complexity multipliers based on all-one polynomial (AOP) and the irreducible equally spaced polynomial [16]. Since the multipliers have been introduced, many bit-serial and bit-parallel low complexity multipliers have been proposed for cryptographic applications. To decrease the computation complexity, Koc and Sunar designed multipliers with a low complexity, which requires m^2 AND gates and m^2-1 XOR gates [6]. In 1997, Fenn et al. presented two bit-serial multipliers using Linear Feedback Shift Register (LFSR) architecture with a low area complexity [15]. Kim et al. presented a bit-serial multiplier, using LFSR architecture based on the inner-product with a very simple hardware [8]. Although Kim et al.'s in [12] and Jeon et al.'s in [9] simplified their hardware complexity; they still require one additional modular reduction after the operation since the results maintained over the extended polynomial basis not over the polynomial basis [1]. The choice of the irreducible polynomial $p(x)$ can play an important role in determining the performance of the implementation of finite field multipliers. For example, using irreducible polynomials with very few non-zero coefficients (e.g., trinomials or pentanomials) can provide significant advantages in terms of both speed

and area. Examples for other special forms of $p(x)$ include all-one polynomials and equally spaced polynomials [7]. However, we do not consider these special polynomials since this paper only focuses on the versatile architectures.

The proposed multipliers can be used as a kernel circuit for exponentiation, inversion, and division architectures.

3 Bit-Serial Multiplication in GF(2m)

Two elements, A(x) and B(x), over GF(2m) can be expressed as polynomials of degree at most m-1 over GF(2):

$$A(x) = a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \ldots + a_1 x + a_0, \quad \text{with} \quad a_i \in GF(2) \quad 0 \le i \le m-1 \tag{1}$$

$$B(x) = b_{m-1}x^{m-1} + b_{m-2}x^{m-2} + \ldots + b_1 x + b_0, \quad \text{with} \quad b_i \in GF(2) \quad 0 \le i \le m-1 \tag{2}$$

We can also specify the field element a(x) in bit-string notation as (a_{m-1} , . . . a_2 , a_1 , a_0). When using polynomial bases representation, the multiplication of field elements $a(x)$, $b(x)$ over $GF(2^m)$ is performed modulo an irreducible polynomial $p(x)$ of degree m over GF(2).

$$P(x) = x^m + p_{m-1}x^{m-1} + p_{m-2}x^{m-2} + \ldots + p_1 x + p_0, \quad \text{with} \quad p_i \in GF(2) \quad 0 \le i \le m-1 \tag{3}$$

Equations (4) to (6) below illustrate the binary (bit-serial) multiplication in $GF(2^m)$. The summation of partial products and the mod $p(x)$ operation are associative and hence they can be carried out in any order.

$$S(x) = A(x).B(x) \bmod P(x) = \left(A(x) \sum_{i=0}^{m-1} b_i x^i \right) \bmod P(x) = \sum_{i=0}^{m-1} A(x).b_i.x^i \bmod P(x) \tag{4}$$

$$S(x) = A(x)b_{m-1}.x^{m-1} \bmod p(x) + A(x)b_{m-2}.x^{m-2} \bmod P(x) + .. + A(x).b_1.x \bmod P(x) + A(x).b_0 \tag{5}$$

$$S(x) = [...[[A(x).b_{m-1}].x \bmod P(x) + A(x).b_{m-2}].x \bmod P(x) + ... + A(x).b_1].x \bmod P(x) + A(x).b_0 \tag{6}$$

The additive and subtractive steps are interleaved, which means the partial products $a(x)$ and $b(x)$ are reduced modulo $p(x)$ before they can be summed up. The interleaved method is described by Equation (5). This equation is rewritten as illustrated in Equation (6) that leads to two different types of realizations. This depends on the order in which the coefficients b_i of the multiplier polynomial $b(x)$ is processed. On the one hand, there is the least significant bit (LSB) first scheme where multiplication

starts with coefficient b_0. But it is also possible to start the multiplication at coefficient b_{m-1} and proceed the multiplier polynomial $b(x)$ in opposite direction, in which this is called the most significant bit (MSB) first scheme. Reference [10] analyzes and compares the hardware requirements, latency and critical path length of LSB/MSB-first multiplier architectures for $GF(2^m)$ [18]. In this paper we concentrate on iterative MSB-first schemes.

3.1 Polynomial Multiplication in $GF(2^m)$ Algorithm

For $k = 1 \ldots m$, the superscript (k) indicates the iteration step, and a subscript i denotes the index of a coefficient.

$$S^0(x) = 0$$
$$S^k(x) = \left(S^{k-1}(x).x + A(x).b_{m-k}\right) \bmod P(x) \tag{7}$$

After m iteration steps, $S^m(x)$ is the final result of $A(x).B(x) \bmod P(x)$. We know that $x^n \bmod P(x) = p_{m-1}x^{m-1} + \ldots + p_1 x + p_0$ then:

$$S^i(x) = \left[s_{m-1}^{i-1}(p_{m-1}x^{m-1} + \ldots + p_1 x + p_0) + s_{m-2}^{i-1}.x^{m-1} + s_{m-3}^{i-1}.x^{m-2} + \ldots + s_0^{i-1}x + b_{m-1-i}A(x)\right] =$$
$$(s_{m-1}^{i-1}.p_{m-1} + s_{m-2}^{i-1} + b_{m-1-i}.a_{m-1}).x^{m-1} + (s_{m-1}^{i-1}.p_{m-2} + s_{m-3}^{i-1} + b_{m-1-i}.a_{m-2}).x^{m-2} +$$
$$+ \ldots + (s_{m-1}^{i-1}.p_1 + s_0^{i-1} + b_{m-1-i}.a_1).x + (s_{m-1}^{i-1}.p_0 + b_{m-1-i}.a_0) = \sum_{j=0}^{m-1} s_i^j x^j \tag{8}$$

Equation (8) describes the functionality of a 1-bit cell that can be used in an MSB-first bit-serial multiplier, where a hardware implementation of the multiplier has been proposed in [17], (Fig. 1). The coefficients of the irreducible polynomial $P(x)$ and the multiplicand polynomial $A(x)$, are stored in $P(i)$ and $A(i)$ registers, respectively. As the coefficients $b(i)$, of multiplier polynomial $B(x)$ are shifted bit by bit in the MSB direction, the partial products $b_i.A(x)$, are produced in the Partial Product component. The LFSR calculates the sum of the $b_i.A(x)$ and also performs the reduction modulo $P(x)$. After m clock cycles, the multiplication product $S(x)$ is produced. This implementation is a MSB-first version since the MSB is the first bit which enters to the multiplier.

4 Proposed Multiplier Architecture

The proposed design for the reconfigurable multiplier that is employed for a variable field degree m is illustrated in Fig. 2. The multiplier employs a parallel structure for driving the global feedback path which is significantly faster than the previously reported techniques.

The proposed hardware implementation consists of a LFSR component that is very similar to the conventional bit-serial multiplier of Fig. 1. It requires 2M extra AND gates and M-1 extra OR gates. Each slice i, consists of two subfield multipliers (AND gates), two subfield adders (XOR gates), two AND gates and 3 one-bit registers.

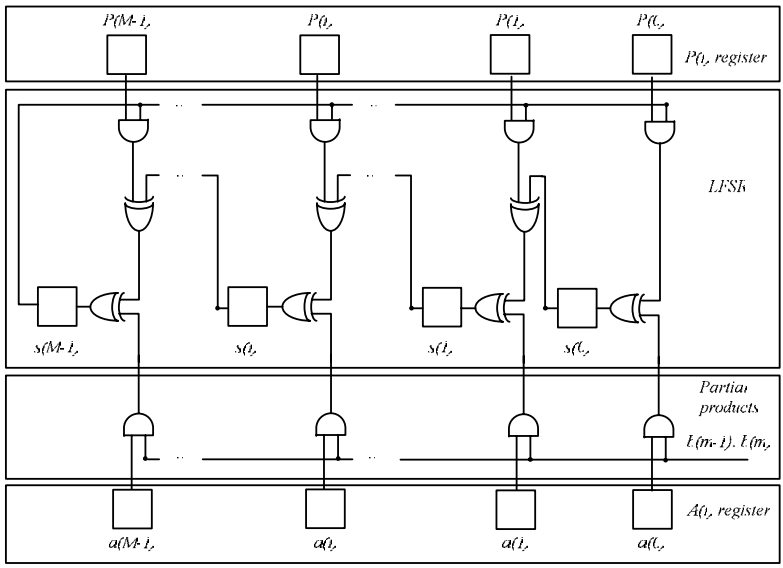


Fig. 1. MSB-first bit-serial multiplier implementation

During computation, each one-bit register s(i) is controlled by means of the corresponding control_A(i) signal. Thus, all the one-bit register s(j), are set inactive by removing their clock signal. So, the unnecessary transitions on all the one-bit register s(j), are eliminated. This gated clock technique results in a significant power dissipation reduction [5]. Each coefficient a(i), of the multiplicand polynomial A(x), is stored in A(i) position of the A(i) register, while each coefficient p(i) of the irreducible polynomial P(x), is stored in the P(i) position of the P(i) register. If an irreducible polynomial of degree m (m < M) is required, the remaining P(j) and A(j) bits of the registers are filled with zeros. The value of each signal control_A(i), is defined as:

$$\text{Control\_A}(i) = \begin{cases} 1 & if \quad i < m \\ 0 & if \quad m \le i \end{cases}$$

The control_B signal by means of a row of AND gates, select one of the one-bit registers s(i). The OR gates (with binary tree structure) transmit selected s(i) into global feedback path. The value of each signal control_B(i), is defined as:

$$\text{Control\_B}(i) = \begin{cases} 1 & if \quad i = m-1 \\ 0 & if \quad i \ne m-1 \end{cases}$$

After m clock cycles, the correct multiplication result is stored in the register s(i). The minimum clock cycle period is determined by the delays of the critical path. It is equal to $2T_{AND}+2T_{XOR}+ (\log_2 m)T_{OR}$, where T_{AND},T_{XOR} and T_{OR} is the delay of the 2-input AND, 2-input XOR, and 2-input OR gates, respectively.

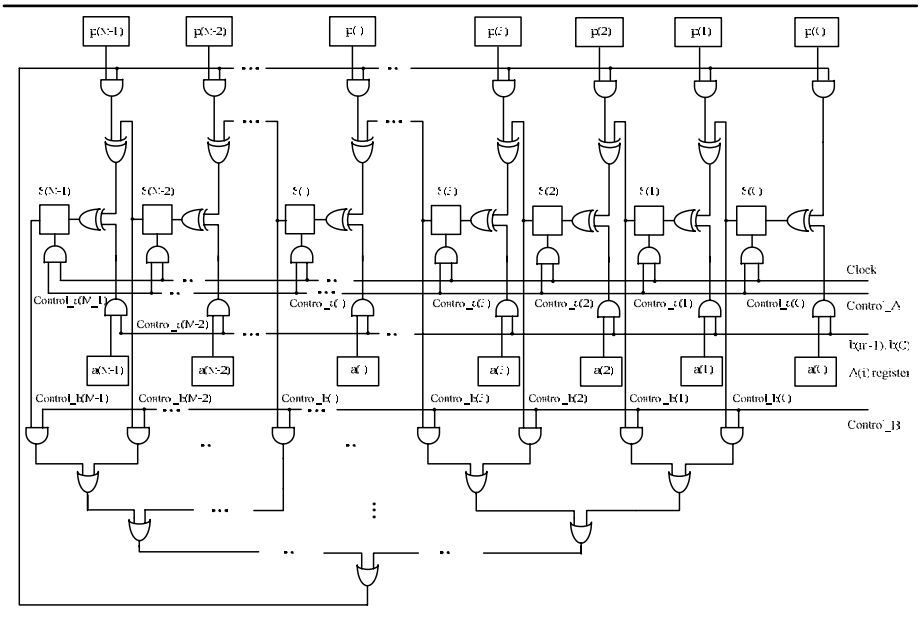


Fig. 2. The proposed reconfigurable bit serial multiplier

Table 1. Area hardware resources and execution time

Multiplier implementation	Ref. [9]	Ref. [15]	Ref. [1]	Ref. [2]	Proposed
# AND	$3m$	$3m$	m+1	$3m$	4m
# XOR	m	$2m$	m+1	$2m$	2m
# REG	m	m^2	$2m$+4	$3m$	3m
# OR	0	0	0	m-1	m-1
# (2:1) EMUX	0	0	0	m	0
# (m:1) MUX	0	M	m+3	0	0
Critical path	$T_{AND}+T_{XOR}$	$3m(T_{XOR}+[\log_2 m](T_{NOT}+T_{AND}+T_{OR}))$	$T_{AND}+mT_{XOR}$	$2T_{AND}+T_{XOR}+T_{NOT}+(m+1)T_{OR}$	$2T_{AND}+2T_{XOR}+[\log_2 m]T_{OR}$
# CLK	m	$3m$	$2m$+1	m	M
†	NO	YES	NO	YES	YES

†: *support any irreducible polynomial*

Table 2. Frequency and multiplication time delay comparisons

Binary field degree (m)	Ref. [2]		Proposed multiplier	
	FPGA frequency (MHz)	Multiplication execution time (μs)	FPGA frequency (MHz)	Multiplication execution time (μs)
106	33.5	3.1	105.04	1.0
119	30	3.9	105.08	1.13
132	26	5	105.11	1.25
158	22.4	7	105.13	1.5
163	22.1	7.4	105.16	1.55
174	19.8	8.8	105.18	1.65
193	18.5	10.4	105.22	1.83
210	17.1	12.3	105.26	1.99

5 Analysis and Comparison

The execution time and the area of the hardware resources for the proposed design are illustrated in Table 1. For the comparison purposes, results from other techniques are also presented. The bit-serial polynomial bases multiplier suggested in [1], is based on the linear feedback shift register using an irreducible All One Polynomial (AOP) over GF(2m). The main disadvantage of this multiplier is the restrictions asserted on the form of the irreducible polynomial. The implementation suggested in [11], is a variable Galois field size multiplier. The multiplication result is computed after 3m clock cycles. The critical path is determined by the multiplexer delay and the delay of the 3-input XOR.

The proposed architecture were modeled with VHDL and implemented in Xilinx Virtex XCV800-4 FPGA device. The FPGA frequency and multiplication execution time for the proposed multiplier implementation are shown in table 2. Comparing the results with method in [2], the implementation of the maximum field degree *M* is 210. The polynomial degree *m* varies from 106 to 210 bits.

6 Conclusion

We presented a versatile architecture for bit-serial multiplication in binary finite fields GF(2m). The multiplier architecture is versatile and has no restrictions on the form of the irreducible polynomial. The clock frequency is immune to changes when the application requires multiplications with variable field sizes. The clock frequency of the implementation is independent of the field size and depends only on the maximum value of field degree M. The main advantage of the proposed architecture is its high order flexibility which allows an easy configuration for variable field size.

References

1. Kim, H.S., Lee, S.W.: LFSR multipliers over GF(2m) defined by all-one polynomial. INTEGRATION, the VLSI journal 40, 473–478 (2007)
2. Kitsos, P., Theodoridis, G., Koufopavlou, O.: An efficient reconfigurable multiplier architecture for Galois field GF(2m). Microelectronics Journal 34, 975–980 (2003)
3. Menezes, A.J., Oorschot, P.C., Vanstone, S.A.: Handbook of Applied Cryptography. CRC Press, Boca Raton (1997)
4. Li, H., Zhang, C.N.: Efficient cellular automata based versatile multiplier for GF(2m). J Inform. Sci. Engng. 18, 479–488 (2002)
5. Chandrakasan, A.P., Brodersen, R.W.: Low power digital CMOS design. Kluwer Academic Publishers, Dordrecht (1995)
6. Koc, C.K., Sunar, B.: Low-complexity bit-parallel canonical and normal basis multipliers for a class of finite fields. IEEE Transactions on Computers 47(3), 353–356 (1998)
7. Lee, C.Y., Lu, E.H., Lee, J.Y.: Bit-parallel systolic multipliers for GF(2m) fields defined by all-one and equally spaced polynomials. IEEE Transactions on Computers 50(5), 385–393 (2001)
8. Kim, H.S., Yoo, K.Y.: AOP arithmetic architectures over GF(2m). Appl. Math. Comput. 58, 7–18 (2004)

9. Jeon, J.C., Kim, H.S., Lee, H.M., Yoo, K.Y.: Bit-serial AB2 multiplier using modified inner product. J. Inform. Sci. Eng. 18, 507–518 (2002)
10. Song, L., Parhi, K.K.: Efficient finite field serial/parallel multiplication. In: Proceedings of the 10th IEEE Int. Conference on Application-Specific Systems, Architectures, and Processors (ASAP 1996), pp. 72–82. IEEE Computer Society Press, Los Alamitos (1996)
11. Hasan, M.A., Ebtedaei, M.: Efficient architectures for computations over variable dimensional Galois field. IEEE Trans. Circuits Syst. I. Fundam. Theory Appl. 45(11) (1998)
12. Koblitz, N.: Elliptic curve cryptosystems. Mathematics of Computation 48(177), 203–209 (1987)
13. National Institute of Standards and Technology (NIST). Digital Signature Standard (DSS), pp. 186–200. Federal information processing standards (FIPS) publication (2000)
14. Lin, S., Costello, D.: Error Control Coding: Fundamentals and Applications. Prentice-Hall, Englewood Cliffs (1983)
15. Fenn, S.T.J., Parker, M.G., Benaissa, M., Tayler, D.: Bit-serial multiplication in GF(2m) using irreducible all-one polynomial. IEE Proc. Comput. Digit. Tech. 144(6), 391–393 (1997)
16. Itoh, T., Tsujii, S.: Structure of parallel multipliers for a class of fields GF(2m). Inform. Comput. 83, 21–40 (1989)
17. Scott, P.A., Travares, S.E., Peppard, L.E.: A fast VLSI multiplier for GF(2m). IEEE J. Sel. Areas Commun. sac-4, 62–65 (1986)
18. Hutter, M., Großschadl, J., Kamendje, G.: A Versatile and Scalable Digit-Serial/Parallel Multiplier Architecture for Finite Fields GF(2m). In: Proceedings of the 4th International Conference on Information Technology: Coding and Computing, pp. 692–700 (2003)

A Nonspeculative Maximally Redundant Signed Digit Adder

Ghassem Jaberipur and Saeid Gorgin

Department of Electrical and Computer Engineering, Shahid Beheshti University and School of Computer Science, Institute for Studies in Theoretical Physics and Mathematics (IPM), Tehran, Iran
{Jaberipur,Gorgin}@sbu.ac.ir

Abstract. Signed digit number systems provide the possibility of constant-time addition, where inter-digit carry propagation is eliminated. Carry-free addition for signed digit number systems is primarily a three-step process. However, the special case of maximally redundant signed digit number systems leads to more efficient carry-free addition. This has been previously achieved by speculative computation of digit-sum values using three parallel adders. We propose an alternative nonspeculative addition scheme that computes the transfer values through a fast combinational logic. The proposed carry-free addition scheme uses a combinational logic, to compute the transfer digit, and the equivalent of two adders. The simulation and synthesis of the two previous works and this work based on 0.13 μm CMOS technology shows that the proposed circuit operates faster and has a lower product of delay × power.

Keywords: Computer arithmetic, Carry-free addition, Signed-digit number system, Maximal redundancy.

1 Introduction

Addition is the basic computer arithmetic operation. Traditional ripple-carry adders are very slow due to the long chain of carry-propagation logic. The latency of an n-digit carry-ripple adder is linearly depending on n (i.e., O (n)) [1]. Carry-accelerating techniques used, for example, in carry look-ahead adders [2] or carry select adders [3] improve the order of latency to O(log n) and O($\sqrt{n}$), respectively. Faster adders are not possible if the sum, as usual, is to be represented in a conventional nonredundant format [4]. However, constant-time (i.e., O(1)) adders may be envisaged if the sum is allowed to be represented in a redundant format [5].

In a radix-r redundant number system, each digit may assume values from a redundant digit set [α, β] whose cardinality is greater than r [6]. The special case of balanced digit set [–α, α] was introduced in the pioneering work of Avizeinis [7] as the signed digit (SD) number systems. The conventional three-step carry-free addition algorithm for SD numbers (see Section 2) has been implemented in hardware, based on different approaches, for improved performance. For example, Fahmy and Flynn offer a maximally redundant SD (MRSD) adder that effectively parallels the three steps of the conventional algorithm [8]. They use three SD adders to speculatively

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 235–242, 2008.

compute three digit-sum values with regards to three different transfer-values. Similar result is reported in [9] based on a different approach for transfer computation.

In this paper, we present a new maximally redundant SD adder that nonspeculatively computes the digit-sums and shows more efficiency compared to previous speculative schemes. Here is a roadmap to the rest of the paper. A background on SD number systems and conventional carry-free addition algorithm is provided in Section 2, the work of [8] and [9] are reviewed in Section 3, we offer, in Section 4, the new nonspeculative addition scheme, Section 5 provides the simulation results, and finally we draw our conclusions in Section 6.

2 Signed Digit Number Systems

In the conventional nonredundant number systems, cardinality ξ of the digit set is equal to the radix r (e.g., the digit set [0, 1] for radix 2 or [0, 9] for radix 10). Notwithstanding the conventional radix-complement or diminished-radix-complement number systems, in order to allow negative numbers, one could think of digit sets with signed values (e.g., radix-5 and radix-16 nonredundant digit sets [–2, 2] and [–8, 7], respectively). Allowing $\xi > r$, leads to redundant number systems, where some values may be redundantly represented by more than one digit combination.

Example 1 (decimal redundancy): Consider the decimal digit set {0, 1,⋯, 9, A, B}, where digits A and B worth 10 and 11, respectively and $\xi = 12 > r = 10$. In this redundant decimal number system, the 3-digit number 110 may also be represented as the 2-digit number AA ($10 \times 10 + 10 = 110$). ◄

The signed digit (SD) number systems, introduced by Avezienis [7], represent a special case of redundant number systems, where the radix-r digit set is $[-\alpha, \alpha]$, and $\alpha \geq r/2$ that leads to $\xi = 2\alpha + 1 > r$. The most useful property of redundant number systems is the possibility of carry-free addition, where the carry propagation chain is limited to a few number of digits [5]. This chain is only one digit long for SD numbers in case of $r \geq 3$ and $\alpha \geq (r+1)/2$ [7], where the carry generated in any position i will not propagate beyond position $i + 1$. The conventional three-step carry-free addition algorithm for SD numbers is presented below as Algorithm 1.

Algorithm 1 (Carry-free SD addition):

Input: Two n-digit radix-r SD numbers $X = x_{n-1}\ldots x_0$ and $Y = y_{n-1}\ldots y_0$, where $-\alpha \leq x_i, y_i \leq \alpha$ for $0 \leq i \leq n-1$.
Output: An $n+1$-digit radix-r SD number $S = s_n\ldots s_0$

I. Compute the n-digit position-sum as a radix-r SD number $P = p_{n-1}\ldots p_0 = X + Y$, by digit-parallel computation of $p_i = x_i + y_i$ for $0 \leq i \leq n-1$.
II. Decompose pi to transfer ti+1 and interim sum wi such that $-\alpha + 1 \leq w_i \leq \alpha - 1$ for $0 \leq i \leq n-1$, $p_i = w_i + r \times t_{i+1}$, and $t_{i+1} = -1$, 0, and 1 for $p_i \leq -\alpha$, $-\alpha < p_i < \alpha$, and $p_i \geq \alpha$, respectively.
III. Compute $s_i = w_i + t_i$, for $0 \leq i \leq n-1$, where no new transfer will be generated by this final addition, and $s_n = t_n$. ◄

Example 2 (Decimal Carry-free addition): Consider the decimal SD digit set [–7, 7], where $\xi = 15 > r = 10$. Fig. 1 illustrates the application of Algorithm 1 on two 4-digit decimal SD numbers. ◄

	i	4	3	2	1	0
	x_i		2	3	−5	4
	y_i		5	6	−6	2
Step I	p_i		7	9	−11	6
Step II	w_i		−3	−1	−1	6
	t_i	1	1	−1	0	
Step III	s_i	1	−2	−2	−1	6

Fig. 1. A decimal SD addition

Each of the Steps I and III of Algorithm 1 contains a digit addition. Also Step II involves a digit comparison of p_i with α. Reducing the number of these three digit operations, if possible, leads to faster carry-free addition. In the next Section two such approaches, which have recently appeared in the literature, are reviewed.

3 Speculative MRSD Addition Schemes

The radix of choice for a redundant number system, besides the special case of radix-10, is practically a power-of-two such that $r = 2^h$. Therefore, given that $\xi > r$, the number of bits for encoding each radix-2^h digit is at least h+1. However, an h+1-bit two's complement encoding of a SD $\in [-\alpha, \alpha]$, allows α to be up to $2^h - 1$ $(\xi = 2\times\alpha+1 \le 2^{h+1} \Rightarrow \alpha \le 2^h - 1)$ corresponding to the maximally redundant SD number system. In this case there is only one invalid h +1-bit digit-value due to -2^h. On the other extreme, $\alpha = 2^{h-1}$ (i.e., the lowest α that meets the Avezienis lower bound $(r+1)/2$), corresponds to the minimally redundant SD number system with $2^h - 1$ invalid h +1-bit digit-value in the intervals $[-2^h, -2^{h-1}-1]$ and $[2^{h-1}+1, 2^h-1]$. One can intuitively expect that the minimum number of invalid h+1-bit digit-values in the maximally redundant case lead to the most efficient SD adder.

Two speculative MRSD adders due to [8] and [9] are depicted in Figs. 2 and 3, respectively. Both schemes use h+1-bit two's complement encoding of the signed digits and rely on triple active hardware redundancy on concurrently computing p_i–1, p_i, and p_i+1 in anticipation of the coming transfer value from position i–1. Fahmy and Flynn observed that the h least significant bits of the interim sum w_i are the same as the corresponding bits of p_i and the MSB of w_i and the transfer bits may be computed by combinational logic. However, the other work is based on comparing p_i with 2^{h-1}, instead of comparison with α in Step II of Algorithm 1. Note that, in Figures 2 and 3

the number of h+1-bit operations in the critical delay path is two and one, respectively. The speculative approach with triple hardware redundancy (i.e., three parallel adders), as followed in these designs, is probably the most straight forward approach. However, we present, in the next section, a nonspeculative MRSD adder based on very fast computation of the transfer digit through a combinational logic.

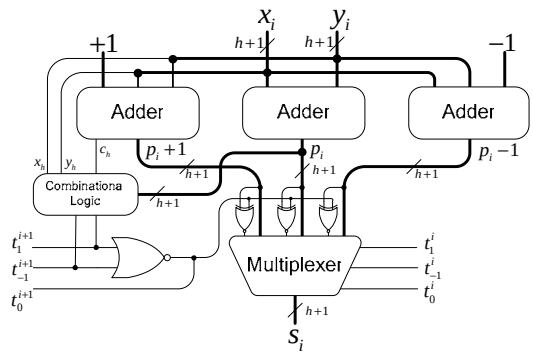


Fig. 2. Position i of MRSD adder of [8]

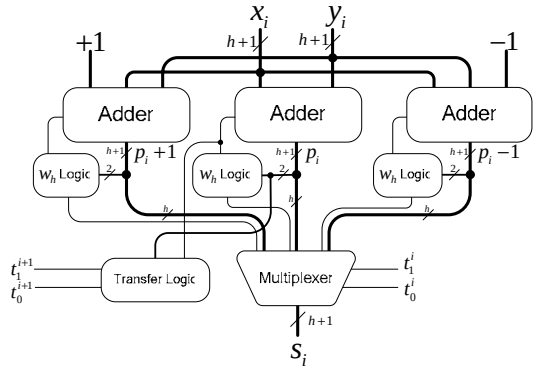


Fig. 3. Position i of MRSD adder of [9]

4 New MRSD Addition Scheme

In SD carry-free addition, the interim sum digit w_i and transfer t_{i+1} can be expressed directly in terms of the radix-2^h digits x_i and y_i (e.g., as functions $w_i = \omega(x_i, y_i)$ and $t_{i+1} = \tau(x_i, y_i)$). However, a direct implementation of these functions in hardware is not practical due to the large number of input bit-variables (i.e., $2\times(h+1)$ bits). Alternatively, and for the sake of efficiency, the strict comparison of p_i and α may be relaxed in the Step II of Algorithm 1. In other words, for a given value of p_i, there may be more than one set of corresponding valid values for w_i and t_{i+1}. Figure 4 illustrates such p_i-values as gray regions, where valid intervals for different transfer values overlap.

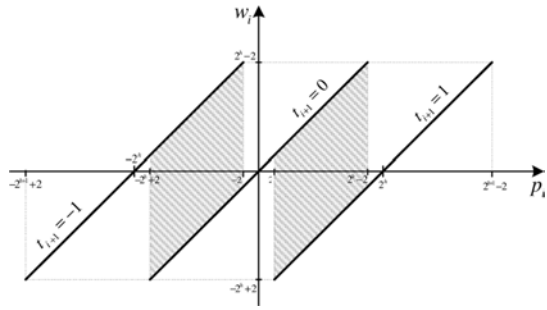


Fig. 4. The overlapping regions of valid values for t_{i+1}

One may take advantage of this imprecision (i.e., choice of different transfer values for a given p_i value), by a careful selection of alternative t_{i+1}-values, to prevent dependency of t_{i+1} on each and all the constituent bits of x_i and y_i. In particular, the case of minimum dependency (i.e., only on X_i^h and Y_i^h, the most significant bits of x_i and y_i) may lead to very fast computation of the transfer. Table 1 shows that t_{i+1} may, indeed, be defined as a function of just X_i^h and Y_i^h, except for five instances of (x_i, y_i) values, where the proposed t_{i+1} leads to an invalid w_i. For example, in the first row of Table 1, $t_{i+1} = 1$ holds for $x_i \geq 0$ and $y_i \geq 0$ except for three cases, where the resultant w_i is not valid (i.e., $w_i \leq -\alpha = -2^h + 1$).

As another example, the strict comparisons of Step II of Algorithm 1 would set t_{i+1}=0 for $-2^h + 1 < p_i < 2^h - 1$. However, the transfer value chosen by Table 1 in some of such cases (e.g., for $x_i = y_i = 1$) is 1.

Table 1. Minimum dependency of t_{i+1} and the exceptions

X_i^h Y_i^h	Range of x_i and y_i	t_{i+1}	w_i	Exceptions on (x_i, y_i) pair
0 0	$x_i \geq 0, y_i \geq 0$	1	$-2^h + p_i$	(0, 0), (0, 1), and (1, 0)
0 1	$x_i \geq 0, y_i < 0$	0	p_i	$(0, -2^h+1)$
1 0	$x_i < 0, y_i \geq 0$	0	p_i	$(-2^h+1, 0)$
1 1	$x_i < 0, y_i < 0$	–1	$2^h + p_i$	None

To take care of the exceptions in Table 1 let $x_i = X_i^h x_i^{h-1} \ldots x_i^0$, $y_i = Y_i^h y_i^{h-1} \ldots y_i^0$, $p_i = P_i^{h+1} p_i^h \ldots p_i^0$, $w_i = W_i^h w_i^{h-1} \ldots w_i^0$ and $t_{i+1} = T_{i+1}^1 t_{i+1}^0$. Equation sets (1) and (2), blow, can be driven from Table 1, where $\varphi = \overline{x_i^{h-1} + \cdots + x_i^1}\ \overline{y_i^{h-1} + \cdots + y_i^1}\ x_i^0 y_i^0$ is a flag indicating the exceptions. Note that x_i, y_i, p_i, w_i, and t_{i+1} are two's complement numbers, where the MSB is distinguished by uppercase variables. Furthermore it is easy to see from Table 1 that $W_i^h = P_i^h$ and $t_{i+1}^0 = 0$ in the two middle rows, $W_i^h = \overline{P_i^h}$ and $t_{i+1}^0 = 1$ in the first and last rows, and the other bits of w_i are the same as corresponding bits of p_i. The reason is that adding $\pm 2^h$ to p_i affects only the bit in position h.

$$T_{i+1}^{1} = X_i^h Y_i^h + \varphi\left(X_i^h + Y_i^h\right), \quad t_{i+1}^{0} = \overline{X_i^h \oplus Y_i^h \oplus \varphi} \tag{1}$$

$$W_i^h = P_i^h \oplus t_{i+1}^{0},\ w_i^j = p_i^j, \text{ for } 0 \le j \le h-1 \tag{2}$$

In spite of the latter improvement regarding the implementation of Algorithm 1, there are still two h +1-bit addition steps; namely $p_i = x_i + y_i$ and $s_i = w_i + t_i$. However, due to the following observations the second addition operation may commence as soon as t_i is available.

- Using Equation-set (1), t_i is available prior to termination of computation of $p_i=x_i + y_i$.
- The bits of w_i, except for the MSB, are available as soon as the corresponding bits of p_i are ready.

Fig. 5 illustrates the new MRSD adder based on the above results. The bold line shows the critical delay path, which passes through exactly h full adder cells as well as a half adder and the transfer logic whose details are depicted in Fig. 6. Note that there is also a combinational logic and a multiplexer in the critical delay path of the two previous designs [8, 9]. Therefore, shorter latency for all values of h, with respect to [8], and moderate values of h (e.g., h = 4), with respect to [9], is expected for the critical delay path of Fig. 5. This is confirmed by the outcomes of simulation and synthesis provided in the next section.

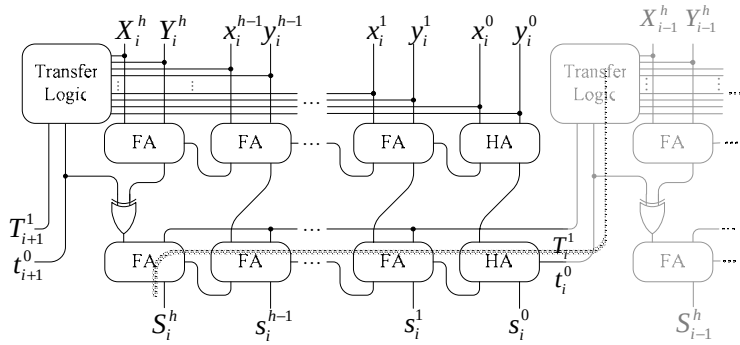


Fig. 5. The new one-step MRSD adder

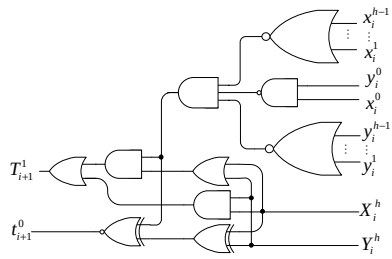


Fig. 6. The transfer logic

5 Results of Simulation and Synthesis

We have implemented the logic of Fig.5 and also those of Figs. 2 and 3, for $h = 4$, in VHDL and run exhaustive tests to ensure correctness. The 5-bit adder in the critical delay path of Fig. 5 and the similar ones in Figs. 2 and 3 are replaced by fast carry look-ahead logic. Subsequently, all the three single digit MRSD adders were synthesized in 0.13 μm CMOS technology. The simulation results are presented in Table 2, where the three designs are compared in terms of latency, power and area. There is also a PDP (product of delay × power) column. All the figures in the bottom row show that the proposed nonspeculative SD adder outperforms the previous works.

Table 2. Simulation results for single digit MRSD adders with h=4

MRSD adder	Delay (ns)	Power (mW)	Area (μm^2)	PDP
Fig.2 [8]	1.30	0.97	985	1.26
Fig. 3 [9]	1.05	1.45	1304	1.52
Fig. 5 (nonspeculative)	0.89	0.77	823	0.68

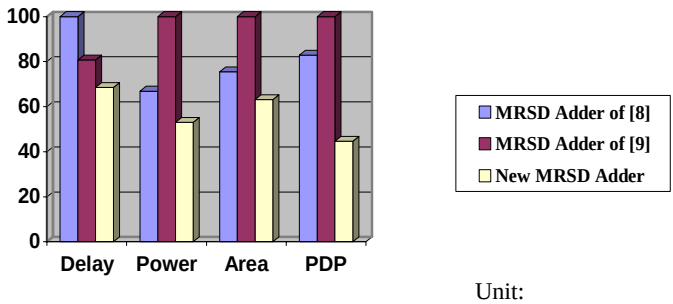


Fig. 7. Illustration of the comparison of three MRSD adders

6 Conclusions

Carry-free addition of radix-2^h signed digit numbers is, in general, a three-step process, where the latency of each step is roughly as long as that of an h +1-bit adder. We propose a nonspeculative maximally redundant radix-2^h signed digit addition scheme with only one h +1-bit digit-adder and an h+1-input combinational logic in the critical delay path. This work is compared with two previously published designs for speculative MRSD adders that use three h+1-bit digit-adders in parallel for improved performance. All the three designs have been simulated by a synthesis tool based on 0.13 μm CMOS technology, where the results are illustrated in Figure 7. Comparing these results with the best of the two previous works, 15.5% less delay is achieved with respect to [9], and 20% less power dissipation, 16.5% less area, and 46% less PDP are the advantages over [8].

Further research is ongoing towards alternative nonspeculative MRSD addition schemes with better performance.

Acknowledgement

This work has been funded in part by grant #600/1627 from Shahid Beheshti University, and in part by grant # CS1386-3-01 from IPM.

References

1. Parhami, B.: Computer Arithmetic: Algorithms and Hardware Designs, Oxford (2000)
2. Doran, R.W.: Variants of an Improved Carry Look-Ahead Adder. IEEE Trans. Computer 37(9), 1110–1113 (1988)
3. Sklansky, J.: Conditional-Sum Addition Logic. IRE Trans. Elec. Comp. 9(2), 226–231 (1960)
4. Winograd, S., Watson, T.J., Heights, Y.: On the Time Required to Perform Addition. Journal of the ACM (JACM) 12(2), 277–285 (1965)
5. Parhami, B.: Generalized Signed-Digit Number System: A Unifying Framework for Redundant Number Representation. IEEE Trans. on Computer 39(1), 89–98 (1990)
6. Jaberipur, G., Parhami, B.: Stored-Transfer Representations with Weighted Digit-Set Encodings for Ultrahigh-Speed Arithmetic. IET Circuits, Devices, and Systems 1(1), 102–110 (2007)
7. Avizienis, A.: Signed-digit number representations for fast parallel arithmetic. IRE Trans. on Electronic Computers EC-10, 389–400 (1961)
8. Fahmy, H., Flynn, M.J.: The Case for a Redundant Format in Floating-point Arithmetic. In: Proc. 16th IEEE Symposium Computer Arithmetic, pp. 95–102. IEEE Computer Society, Los Alamitos (2003)
9. Jaberipur, G., Ghodsi, M.: High Radix Signed Digit Number Systems: Representation Paradigms. Scientia Iranica 10(4), 383–391 (2003)

System-Level Assertion-Based Performance Verification for Embedded Systems

Hassan Hatefi-Ardakani, Amir Masoud Gharehbaghi, and Shaahin Hessabi

Department of Computer Engineering Sharif University of Technology
Tehran, Iran
{hatefi,gharebaghi}@ce.sharif.edu, hessabi@sharif.edu

Abstract. As contemporary digital systems specifically embedded systems become more and more complex, taking advantage of system-level design is being more widespread. Many embedded systems must operate under strict timing constraints. One of the best methods for examining timing constraints in an embedded system can be done via the performance verification. In this paper an assertion-based verification methodology has been proposed for verifying system-level timing constraints in an embedded system. Performance assertions are specified by an assertion language at the transaction-level of abstraction. A Turing machine and a structure named performance evaluator have been coupled to provide a computational model for a performance assertion. We have developed a tool that can automatically generate a C++ code from input assertions. The result code operates as the computational model and checks the assertions by applying a simulation-based trace analysis approach. Through a case study, we demonstrate usefulness and effectiveness of our methodology.

Keywords: Assertion-Based Verification, Computational Model, Performance Verification, Transaction-Level of Abstraction, Performance Assertions.

1 Introduction

Nowadays, the increasing complexity of electronic systems demands a more sophisticated design and test methodology. As the complexity of systems increases, exclusively designing at Register Transfer Level (RTL) is not effective anymore; as a result, designers need to start from higher levels of abstraction. The increase in the system complexity intensifies the existing gap between design and verification technique capacities. On the other hand, a large number of today's systems are real-time in nature. Verification of these systems requires satisfaction of some performance and timing constraints. Therefore, a new verification methodology must be developed and adapted for today's system-level designs which address the problem of the system-level performance and timing verification.

Assertion-based verification is a modern, powerful verification paradigm. With the assertion-based verification, assertions are used to capture the required behavior of the design. The design then can be verified with regard to those assertions using simulation and/or static verification (e.g. model checking) techniques. By inserting embedded assertions into HDL (Hardware Description Language) descriptions, designers

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 243–250, 2008.

can validate their designs. Property Specification Language (PSL) [1] and Synopsys OpenVera [2] are two instances of a most famous assertion language. These languages are suitable for the functional verification but not the performance validation. They are essentially adapted to RTL, but the verification of transaction-level designs necessitates some preliminary concepts and requirements as described in [16]. Logic Of Constraints (LOC) [3,4,5] provides a formal logic to specify high-level quantitative performance and functional constraints. Constraints are specified at the transaction-level of abstraction and checked by using a simulation-based trace analysis approach [6]. Although, LOC can be applied to an embedded system described at transaction-level of abstraction, in comparison with our work, it has presented a different mathematical analysis for performance assertions. Deriving low-level timing constraints from high-level timing constraints are presented in [7,8]. The proposed mechanism can be established by two automation tools RADHA [9] and RATAN [10]. The method provides the ability of analyzing high-level temporal constraints and synthesizing them to lower level of abstraction. However, it is not suitable for validation of timing constrains. Regardless of it, in this method, timing constraints are limited to simple formulas such as rate and latency.

In our pervious work [15], we have presented a methodology for functional and performance verification of transaction-level designs. To apply a verification approach to system-level designs, in this paper, we have presented a transaction-level verification methodology for validating timing assertions specified by a PSL-like assertion language. An offline simulation-based trace checking approach is proposed to efficiently verify assertions. A tool has been developed for automatically generating the trace checker from our assertions. A computational model for evaluating performance assertions is presented here. Using a Turing machine and a performance evaluator, the assertion is checked on a sample simulation trace of the design. The difference between our method and others is the mathematically accurate definition of assertions and terms related to them based on our computational model.

This paper presents two main contributions. The first contribution is to deploy the PSL-like syntax for verifying performance assertions of the design at the transaction-level of abstraction. Introducing a computation model using the Turing machine idea is our second contribution. To demonstrate the efficiency of our methodology, a JPEG decoder has been selected as the case study. Several assertions have been specified and checked using our verification approach.

In the next section, we introduce our verification language. Section 3, depicts our computational model. In Section 4, we specify our assertion checking approach and an automation tool for it. Experimental results are discussed in Section 5 and Section 6 is our conclusions and future directions.

2 The Assertion Language

In this section, we will explain the assertion language. Basic concepts of our methodology will be discussed at the first. Afterwards, the language syntax and semantics will be described.

2.1 Overview

The assertion-based verification gives designers the ability of defining properties of a system. The core of the method is based on the assertion which describes required behaviors of the design under verification. The assertion can be formally specified using an assertion language. The main part of an assertion is a *property*. It is made up of a Timing Condition (TC) and an operator that controls it. In a property, we are able to define and use events and their related values (mostly time). At the transaction-level, a sequence of event occurrences constructs the execution trace of the system. Each event occurrence can change the state of the design under execution. Therefore, a property must be evaluated on a sequence of state transitions along the time that we name it the *path*.

We define the path $p=s_0s_1...s_{n-1}$ on the time sequence $T_p=t_0t_1...t_{n-1}$, where, s_i is the state of the design at time t_i such that $t_i < t_{i+1}$, (i =0,1...,n-2). Mathematically, s_i is the set of all events happened at t_i, hence $s_i \subset \sum$, where $\sum$ denotes the set of all events which can occur in the design. A TC can be evaluated according to the three-value logic. On each state of a path, the value of a TC can be *true* or *false* or *unknown*. In case that all of events associated with a TC have occurred, the TC will be evaluated to *true* or *false*, otherwise it will be *unknown*. We will define two important terms, *hold* and *hold tightly* as the criterion of the property matching.

2.2 The Syntax and Semantics

Performance assertions can define timing behavior of the system into a TC. The TC associates the time and the value of signals in the design with event occurrences. Therefore, a TC must be evaluated on the sequence of event occurrences along the time (i.e. the path). We claim that performance assertions can be used to specify very common and useful timing behaviors:

- *Rate*: the delay between consecutive occurrences of *e* is depicted in (1). It means that the event *e* must be occurred every 10 time units (*time* is the timing variable);

  assert always time@(***next_p***[1] e) – time@(***next_p***[0] e) == 10; (1)

- *Latency*: according to (2), the latency between *e1* and *e2* has been limited to at most 30 time units. Note that, times of *i*th occurrence of *e1* and *e2* are extracted from the path for evaluating this assertion;

  assert always time@(***next_p***[0] e1) – time@(***next_p***[0] e2) <= 30; (2)

- *Throughput*: the delay between each occurrence of *e* and 100th next occurrences of it is forced to be less than 1000 as shown in (3).

  assert always time@(***next_p***[100] e) – time@(***next_p***[0] e) < 1000; (3)

Using Boolean operators and this formalism, it is possible to define various useful performance properties.

An assertion is composed of *assert* directive followed by a performance property. A performance property contains a TC preceded by an operator (e.g. *always*). For example, the TC of (3) is time@(***next_p***[100] e) – time@(***next_p***[0] e) < 1000. A TC consists of expressions like *exp*@(*a*) and simple arithmetic and logic operators.

exp@(*a*) denotes the value of *exp* at occurrences of *a*, where *exp* is the time or cycle of design or any other signal value in the system. The *next_p* operator is located before an event and inserts an implicit index variable identified by *i* to the TC. The constant after *next_p* is always used as the offset of the index variable. For instance, the semantics of (3) can be illustrated by: For $i > 0$, **time_of**(*e*[*i*+100]) – **time_of**(*e*[*i*]) <1000, where *e*[*n*] means *n*th event occurrence of *e*.

For each TC, we must implicitly define the variable *i* to determine which number of event occurrences are needed for evaluating TC and then, to detect on which prefix of the path, the TC is computable. By adding the value of *i* to the constant integer preceded by *next_p* operator, the number of each event occurrences, to assess the assertion, are determined. For example, in the following TC:

$$\text{time@}(\textbf{\textit{next\_p}}[2]\ a) - \text{time@}(\textbf{\textit{next\_p}}[0]\ b) < 3; \quad (4)$$

For *i*=1, time at the third occurrence of *a* minus time at the first occurrence of *b* is compared with 3. Therefore, each prefix of the path that contains at least three occurrences of *a* and at least one occurrences of *b* can be used to evaluate (4) for *i*=1. We identify this class of prefix as the computable prefix of thepath for *i*=1. The value of *i* is initialized to one and increases until the path is entirely processed. A sample of execution trace on a path is shown in Table 1. For *i*=1, the computable prefix is up to state 4, and the TC is evaluated to *false* on this state. For *i*=2, fourth occurrence of *a* (on state 6) and second occurrence of *b* (on state 5) are used. Then, the TC is evaluated to *true* on state 6. On other states, all of information required for evaluating the TC is not ready; as a result, its value will be *unknown*.

Table 1. Evaluation of TC (4) on a sample path containing 8 states (F=False, T=True, U=Unknown)

State	0	1	2	3	4	5	6	7
Time	1	3	4	5	6	7	9	10
A	*T*	*F*	*T*	*F*	*T*	*F*	*T*	*F*
B	*F*	*T*	*F*	*F*	*F*	*T*	*F*	*F*
I	1	1	1	1	1	2	2	3
Value of (4)	*U*	*U*	*U*	*U*	*F*	*U*	*T*	*U*

For each value of *i*, there is a computable prefix of a path to evaluate a TC. As a result, the TC implies a class of computable prefixes described by a language on a path. For example, the language of (4) with variable *i* on the path *p* is:

{x∈prefix(p): x is ended to *a*, and contains i+2 occurrences of *a* and at least i occurrence(s) of *b*} ∪ {x∈prefix(p): x is ended to *b*, and contains at least i+2 occurrences of *a* and i occurrence(s) of *b*}

Where *i*=1,2,...etc and *prefix*(*p*) is the set of all prefixes of *p*. To compute a TC, the first step is detecting the computable prefix for current value of *i*. It can be done using automata that accept the language related to the TC. In our methodology, the language associated with a TC may be context free or context sensitive, so its accepting automata can not be a Finite State Machine (FSM). For instance, the previous language is context free, and then it can not be accepted by an FSM. Therefore the accepting machine of our methodology will be more complicated than an FSM.

3 Performance Evaluation Model

In this section, we will introduce the computational model for assessing a TC. Then, the semantics of our assertion language will be described based on the defined computational model.

3.1 Computational Model

A TC is computed using a Turing machine, and a performance evaluator which must cooperate in the same work space. Turing machine M is a 5-tuple $M=(Q, \Sigma,\delta,q_0,F)$, where:

- Q = set of internal states of M;
- Σ = input alphabet for M, including the *blank symbol* #, all events defined in the design and some replacing symbol described later;
- δ: $Q\times 2^{\Sigma}\rightarrow Q\times\Sigma\times\{L,R\}$, is the *transition function*;
- $q_0 \in Q$, is the starting state;
- $F \subset Q$, is the final states set.

Turing machine M should accept the computable prefix of the path for current value of i. For TC (4), M accepts prefix $s_0s_1...s_4$ for i = 1, and prefix $s_0s_1...s_6$ for i = 2. The evaluation of a TC via the processing of computable prefixes is assigned to the performance evaluator. Performance evaluator PE is a 5-tuple $PE=(Q, \Sigma,A,C,p)$, where:

- Q is the internal states of M that cooperate with PE;
- Σ is the input alphabet of M that cooperate with PE;
- A is the set of actions that PE must done at the state transitions of M (A is called *action set*);
- $C:Q\times 2^{\Sigma} \rightarrow A \times \{true, false, unknown\}$, is *computation function*;
- $p(x_1,x_2,...,x_n)\rightarrow\{true,false,unknown\}$, is *property function*, that $x_1,x_2,...,x_n$ are property variables.

The satisfaction level of a TC is calculated by property function p that is a three-value logic function obtained from the TC. For example, Expression (5) can be considered as the property function of (4). The computation function C must determine the required action for calculation of p. Also, the satisfaction level of the TC on each state of a path should be gotten out by C, according to the result of p and the internal state of M. C is synchronous with transition function δ in M.

$$p(x_1,x_2) = ((x_1 - x_2) < 3) \tag{5}$$

PE is responsible for evaluating TC based on the state of M. When M is working, required values for estimating the property function (i.e. $x_1,x_2,...,x_n$ in $p(x_1,x_2,...,x_n)$) are gathered by the PE. This task is accomplished by selecting appropriate actions from A and doing them in computation function C. Then, when a path is accepted by M, the result of $p(x_1,x_2,...,x_n)$ is considered as the value of the TC.

4 Computer-Aided Assertion Checking

In this section, we present a simulation-based trace-analysis approach that is used to validate our assertions in an effective and applicable way. General verification flow in

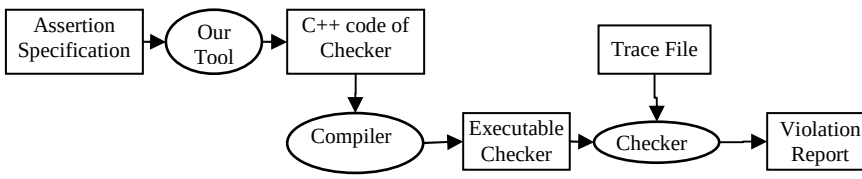


Fig. 2. General methodology for computer-aided verification

the methodology is depicted in Fig. 1. It consists of two levels: automatic checker generation and checking of trace file. The former is accomplished by our tool: an automatic checker generator used for creating a C++ source of the checker. In the second level, the source code is compiled into an executable that examines the simulation trace and reports any constraint violation. To do this, the tool creates C++ source of the checker from input assertions. It is composed of three main components: a lexical analyzer, an assertion compiler and a code generator. The code generator is the main part of our tool. The evaluation algorithm the code generator must be implemented for each assertion is essentially based on its computational model. The tool can create the C++ description of M and PE as the evaluation engine of each assertion. The tape of M and PE is realized using several queues, one for each event involved in the assertion. To address the unbounded memory of the tape, a memory management mechanism is embedded into the trace checker. The mechanism periodically investigates queues to determine which events are processed and not required for evaluation algorithm, so it releases them.

5 Experimental Result

In this section, we use the assertion language described in Section 2, to verify a JPEG decoder design. The JPEG decoder uses ODYSSEY [11] as its synthesis methodology. It is an object-oriented C++ code that is synthesized into a SystemC description using the ODYSSEY tool [12]. We simulate this SystemC description in order to generate execution trace of the system. The assertions are planned to verify performance properties of JPEG decoder. Full details of the JPEG decoder design are presented in [13]. The execution trace of the design is reported into a simulation log file. The design is object-oriented in essence, therefore we have planned assertions somehow they can validate the relationship among objects, methods and data members. We can designate the assertions to examine method execution time, method call rate and the dependency between execution time and data flow of methods. Assertions are verified based on the mechanism described in Section 4 using an offline trace checking approach.

We have specified three performance assertions as depicted in Table 2. In Assertion A1, the execution time of the method *read_8_bits*() are examined by two events *read_8_bits_call* and *read_8_bits_return*. *cycle* denotes the simulation cycle of the design. The average of *read_16huff_bits* call for each 50 invocations is checked by Assertion A2. Finally, Assertion A3 specifies that if the return value of *read_16_huff_bits* is less than 1000, its execution time of must be less than 20.

Table 1. Assertions that is used to verify the JPEG decoder

A1	***assert always*** *cycle*@(***next_p***[0]read_8_bits_call)-*cycle*@(***next_p***[0] read_8_bits_return) <12
A2	***assert always*** (*cycle*@(***next_p*** [50]read_16huff_bits_call) - *cycle*@(***next_p***[0] read_16huff_bits_call))/50 < 200
A3	***assert always*** *return*@(***next_p*** [0]read_16huff_bits_return)<1000->*cycle*@(***next_p***[0] read_16huff_bits_return))- *cycle*@(***next_p***[0] read_16huff_bits_call))<20

We have generated the checker for each assertion described in Table 2. The design has been simulated during the decoding of three different JPEG files and makes three different trace file. The checker has been applied to these trace files on our machine and then its memory and CPU usages during verification is reported in Table 3. The effect of the trace line number on the CPU time is more than the effect of the assertion structure. Unlike, for memory usage the latter is more noticeable. The trace checkers utilize small amount of CPU and memory and therefore they can be effectively used for the performance verification of system-level designs.

Table 2. Memory and CPU usage of trace checker for verifying assertions of Table 2

Lines of Trace		**175400**	**877000**	**3508000**
A1	**Resident Memory (KB)**	880	880	880
	CPU Time (ms)	810	2290	8410
A2	**Resident Memory (KB)**	872	872	872
	CPU Time (ms)	880	2470	8900
A3	**Resident Memory (KB)**	884	884	884
	CPU Time (ms)	1150	3180	12170

6 Conclusions

In this paper, we presented an assertion-based verification methodology that is suitable for validating performance formulas in system-level designs. We introduced an assertion language for specifying performance properties. We illustrated the ability of the assertion language to specify very common and useful performance properties. Then the semantics of our assertion language was described. A computational model using a Turing machine and a property evaluator was proposed to precisely define semantics of an assertion. Also, we showed that our tool can automatically generate the trace checker based on the computational model from input assertions. Our methodology was used for verification of a JPEG decoder where several assertions were written, and checked on the simulation trace to demonstrate its usefulness and effectiveness.

We are planning to extend our assertion language to support more useful operators. Also, we are intending to utilize the computational model for the formal verification of timing constraints.

References

1. Property Specification Language (PSL) Reference Manual, http://www.eda.org
2. OpenVera Assertions White Paper. Synopsys, Inc., Mountain View (2002)
3. Balarin, F., Watanabe, Y., Burch, J., Lavagno, L., Passerone, R., Sangiovanni-Vincentelli, A.: Constraints specification at higher levels of abstraction. In: International Workshop on High Level Design Validation and Test, Monterey, CA, USA, pp. 129–133 (2001)
4. Chen, X., Hsieh, H., Balarin, F., Watanabe, Y.: Verifying LOC based functional and performance constraints. In: International Workshop on High Level Design Validation and Test, San Francisco, CA, USA, pp. 83–88 (2003)
5. Chen, X., Hsieh, H., Balarin, F., Watanabe, Y.: Logic of constraints: a quantitative performance and functional constraint formalism. IEEE Trans. on CAD of Integrated Circuits and Systems 23(8), 1243–1255 (2004)
6. Chen, X., Hsieh, H., Balarin, F., Watanabe, Y.: Automatic trace analysis for logic of constraints. In: 40th Design Automation Conference, Anaheim, CA, USA, pp. 460–465 (2003)
7. Ramanathan, D., Jejurikar, R., Gupta, R.K.: Timing driven co-design of networked embedded systems. In: Proceedings of the 2000 conference on Asia South Pacific design automation, Yokohama, Japan, pp. 117–122 (2000)
8. Dasdan, A., Ramanathan, D., Gupta, R.K.: A timing driven design and validation methodology for embedded real-time systems. ACM Trans. on Design Automation of Electronic Systems 3(2) (1998)
9. Dasdan, A., Ramanathan, D., Gupta, R.K.: Rate derivation and its applications to reactive real-time embedded systems. In: 35th Design automation Conference, pp. 263–268 (1998)
10. Dasdan, A., Mathur, A., Gupta, R.K.: RATAN: A tool for rate analysis and rate constraint debugging for embedded systems. In: Euro. Design and Test Conf., pp. 2–6. IEEE Press, Los Alamitos (1997)
11. Goudarzi, M., Hessabi, S., Mycroft, A.: Object-oriented ASIP design and synthesis. In: Forum on Design & Specification Languages (FDL), Germany (2003)
12. Goudarzi, M., Hessabi, S.: The ODYSSEY tool-set for system-level synthesis of object-oriented models Embedded Computer Systems. In: Hämäläinen, T.D., Pimentel, A.D., Takala, J., Vassiliadis, S. (eds.) SAMOS 2005, vol. 3553, pp. 394–403. Springer, Heidelberg (2005)
13. MohammadZadeh, N., Hessabi, S., Goudarzi, M.: Software Implementation of MPEG2 Decoder on an ASIP JPEG Processor. In: International Conference on Microelectronics (ICM 2005). IEEE Press, Islamabad (2005)
14. Ecker, W., Esen, V., Velten, M., Hull, M.: Requirements and Concepts for Transaction Level Assertions. In: 24th International Conference on Computer Design (ICCD 2006), San Jose, California (2006)
15. Hatefi-Ardakani, H., Gharebaghi, A.M., Hessabi, S.: A Performance and Functional Assertion-Based Verification Methodology at Transaction-Level. In: International Conference on Microelectronics (ICM 2007). IEEE Press, Egypt (2007)

The Effect of Core Number and Core Diversity on Power and Performance in Multicore Processors

A. Zolfaghari Jooya and M. Soryani

Computer Science Dep., Iran University of Science and Technology,
Tehran, Iran
al_zolfaghari@comp.iust.ac.ir,
soryani@iust.ac.ir

Abstract. Today, multi-core processors dominate server, desktop and notebook computer's market. Such processors have been able to decrease power consumption and thermal challenges that designers face within single core processors. In order to improve multi-core processor's performance, designers should choose the best set of cores based on power consumption and execution delay. In this paper, we study several architectures that are composed of a configurable number of cores. We use three cores with different levels of performance and power consumption. Then, we implement different configurations of a multi-core processor. In each configuration, which has a different set of cores, we run benchmarks with various numbers of simultaneous threads, from 1 up to 32. Power consumption and execution delay of each configuration has been measured. It has been shown that the best configuration is a heterogeneous multi-core processor that is composed of 16 cores in our bounded area. Then, we examined various ways that threads can be assigned to different cores in the best configuration. It is shown that for serial workloads the best choice is to use high performance cores, but in parallel workloads that consist of multiple threads, a mixture of cores with different performance levels gives the best performance.

Keywords: Heterogeneous multi-core processor, power consumption, execution delay, multithreaded benchmark.

1 Introduction

The constant decrease in feature size leads to an increase on transistor count on chip area which enables processor architects to improve performance by designing more complex processors. This amount of transistors on the chip area increases power consumption and produces more heat. These two factors, i.e. power consumption and thermal management, are the most important design limitation factors today [7,8,9].

From computational point of view, we have already extracted the easy ILP (instruction-level parallelism) through techniques like superscalar processing, out-of-order processing, etc.. The ILP that is left is difficult to exploit. However, technology keeps making transistors available to us at the rate predicted by Moore's Law [11]. We have reached a point where we have more transistors available than we know how to make effective use of in a conventional monolithic processor environment.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 251–258, 2008.

There is diversity in workloads that a typical processor is expected to run. This diversity can be due to diversity among applications or different threads of the same application. It can also be due to diversity across varying program phases within an application or varying processor load. Instead of using all the transistors to construct a monolithic processor targeting high single-thread performance, we can use the transistors to construct multiple simpler cores where each core can execute a program (or a thread of execution) [1]. A multicore processor provides increased total computational capability on a single chip without requiring a complex microarchitecture.

As a result, simple multicore processors have better performance per watt and area characteristics than complex single-core processors [10]. Multicore processors have better adaptability whit workloads and tune the number of active cores according to different workloads or different phases of a single application. This adaptability leads to consume less power and reduce temperature of the chip.

In this paper we implement three cores, each with a different level of performance, power and area. We use different composition of these cores on our bounded area and show delay and power level of each configuration. After finding the best composition of cores, we study the effect of different ways that threads can be assigned to different cores.

Many works have appeared in the literature exploring the design space of multicore processors from the point of view of different metrics and application domains.

Huh et al. [2] evaluate the impact of several design factors on the performance. The authors discuss the interactions of core complexity, cache hierarchy, and available off-chip bandwidth. The paper focuses on a workload composed of single-threaded applications. It is shown that out-of-order cores are more effective than in-order ones.

In [3], Kumar et al. propose a single-ISA heterogeneous multi-core architecture as a mechanism to reduce processor power dissipation. They assume a single chip containing a set of diverse cores that target different performance levels and consume different levels of power. They describe an example architecture with five cores of varying performance and complexity.

In [4], Jouppi et al. demonstrate that single-ISA heterogeneous multi-core architecture can provide significantly higher performance in the same area than a conventional multiprocessor chip. It does so by matching the various jobs of a diverse workload to the various cores. Authors show that this type of architecture covers a spectrum of workloads particularly well, providing high single-thread performance when thread parallelism is low, and high throughput when thread parallelism is high.

In [5] Monchiero et al. target an architecture composed of a configurable number of cores, a memory hierarchy consisting of private L1 and L2, and a shared bus interconnect. They explore the design space varying the number of cores, L2 cache size and processor complexity, showing the behavior of the different configurations/applications with respect to performance, energy consumption and temperature.

The rest of paper is organized as follows. In section 2 we discuss our design methodology and define the details of microarchitecture, simulator and benchmarks. In section 3 we introduce simulation results. Finally, Section 4 concludes the paper.

2 Methodology

2.1 Microarchitecture

We have implemented three cores with different levels of performance, power and different chip areas. Our cores are similar to Alpha processors (alpha21064 [12], alpha21164 [13] and alpha21264 [14]). The core characteristics are similar to the ones used in [3]. Table 1 summarizes these characteristics. We have changed the issue-width of EV5 from 2 to 4 and its instruction and data cache size from 8 KB to 16 KB, in order to increase performance distance between EV4 and EV5. More than half of the chip area was considered for L2 cache and interconnections. In the remainder of chip area we can put four EV6 (alpha21264), or twenty EV5 (alpha21164), or forty EV4 (alpha21064), or a different mixture of these cores. We consider that all cores are implemented in 100 nm technology and run at 2.1 GHz.

A large L2 cache (4 MB) was used to be shared between cores. The L2 cache is a 4 way set associative with a block size of 32 bytes.

Table 1. Characteristics of the cores

core	EV4	EV5	EV6
Issue-width	2 (in-order)	4 (in-order)	6 (OOO)
I-cache	8 KB , DM	16 KB , 2 way	64 KB , 4way
D-cache	8 KB , DM	16 KB , 2 way	64 KB , 4way
B-predictor	static	hybrid	hybrid
Area (mm2)	2.5	5	25
Power (watt)	5	7.5	20

2.2 Simulator and Benchmarks

We used the SESC [15] simulator which is a cycle accurate architectural simulator. It models a very wide set of architectures such as single processors, CMPs and thread level speculation. For simulating heterogeneous multi-core processors, we modified the configuration file and added different core configurations.

We have used four scientific/technical parallel workloads from splash2 [6]. These workloads consist of two applications and two computational kernels. The kernels are FFT and LU decomposition. The two applications that we have used are Barnes and Ocean. Table 2 lists the benchmarks that we selected and the input parameters of each one.

3 Experimental Results

3.1 Different Compositions of Cores

In this section we present the simulation results for different core configurations of our processor. We show execution delay (msec), power and energy-delay of our

simulation results. The first part of Fig. 1 shows the execution delay for some configurations of the processor for ocean benchmark.

Note that 2.6.8 means our processor has two EV6, six EV5, eight EV4 cores. EV6 and EV4 are the highest performance and the lowest performance cores respectively. Also there are some assumptions in our simulation:

Number of simultaneous threads is considered to be power of two.

Number of simultaneous threads that run on cores must be less than or equal to the number of cores.

If the number of simultaneous threads is less than the number of cores, we assume that extra cores are off and don't consume power.

The power that is consumed by cores is reported in the results, not total chip power.

We have used static scheduling. The cores with higher performance have priority for usage to cores with lower performance (when threads are assigned to cores), so threads are first assigned to high performance cores and then, if there are any left, they are assigned to other cores i.e. if we run 8 threads on 2.6.8 multi-core processor, 2 threads are assigned to two EV6 and 6 of them run on six EV5.

Table 2. Benchmarks and input parameters

Bench.	Description	input
FFT	Perform 1D fast Fourier transform using six-step FFT method	m = 16 l = 5 n = 1024
Ocean	This application studies the role of eddy and boundary currents in influencing large-scale ocean movements.	N = 130
LU	Parallel dense blocked LU factorization	n = 1000 b = 64
Barnes	implements the Barnes-Hut method to simulate the interaction of a system of bodies	nbody = 64 seed = 45 fleaves = 5.0

There are three homogeneous multi-core processors in this figure (e.g. 4.0.0 that consists of four EV6 cores, 0.20.0 that consists of twenty EV5 cores and 0.0.40 that consists of forty EV4 cores). Other configurations implement heterogeneous multi-core processors with two or three different cores.

Note that our simulation is limited with this fact that the number of workload's threads must be power of 2 and less than the number of cores that we implement in our design. Therefore we can't show execution delay of 2.6.8 configuration, which has 16 cores, for the benchmark that has 32 simultaneous threads. But we can see that 1.0.31 configuration has the best execution delay for 32thread- benchmark in all figures. If we run a serial workload on a low performance core, we will get the worst delay.

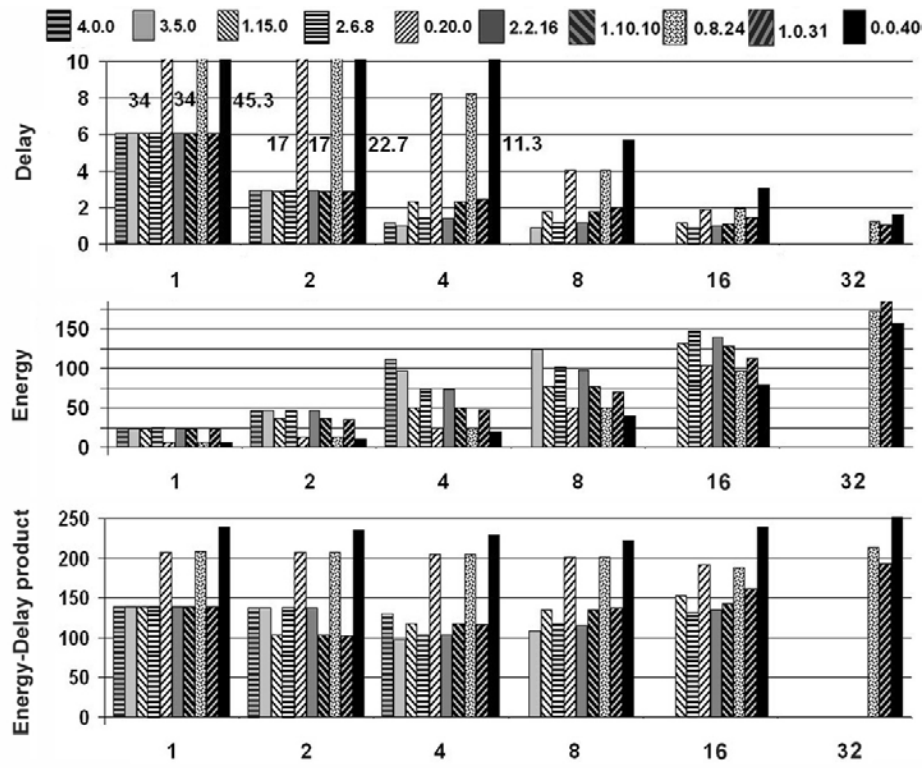


Fig. 1. Execution-Delay, energy and energy-delay product for ocean benchmark for different thread numbers

The middle part of Fig. 1 shows the amount of power that is consumed in different configurations for ocean benchmark. It is clear that if we run the application's threads on high performance cores, we must pay the penalty that is more power consumption. Note that the total power consumption is increased with the number of simultaneous threads, because more cores are used simultaneously.

The last part of this figure shows Energy-Delay product diagrams. It can be seen that 0.0.40 (homogeneous multi-core) always has the worst Energy-Delay curve. The best configurations are 2.6.8 and 1.0.31 for workloads that have 32 simultaneous threads.

Depending on the goal of the design; the scheduler can use selected cores to reach the best result. For example consider that we implement the 2.6.8 configuration for multi-core processor in an embedded system and the amount of power that is consumed is critical. Scheduler can run multithreaded workload on the low performance cores that consume less power. If we run 8 threads on 8 EV4 cores in 2.6.8 configuration, the power that is consumed is 1/3 of when we run 8 threads on 2 EV6 and 6 EV5 cores. If the goal of design is to reach the best execution delay, the threads must run on high performance cores.

Heterogeneous multi-core processors are suitable for systems that have variable goals in their life time. For example embedded systems can have multiple goals, depending on environmental or operational situations. Reducing power or execution delay or both of them can be such system's goals. In these systems heterogeneous multi-core processors have the best compatibility with these goals, and scheduler can decide that which cores be used to run the threads.

We implemented many other possible configurations. We found that 2.6.8 configuration still has the best performance and some other configurations have performance near to that. For example 2.4.12 and 2.5.10 have performances near 2.6.8 and 3.5.0 has the best Energy-Delay product. Other benchmarks shown in Table 2 have also been used in the experiments and similar results have been achieved.

3.2 Thread Assignment Policy

In this section we choose the configuration that had the best performance in previous section, and study the effect of different thread assignment policies. We ran different number of simultaneous threads (from 1 thread to 8 threads) on different cores and compared the execution delay, power and energy-delay of each simulation result. We found the best assignment for each number of threads from the execution delay, power and energy-delay aspects.

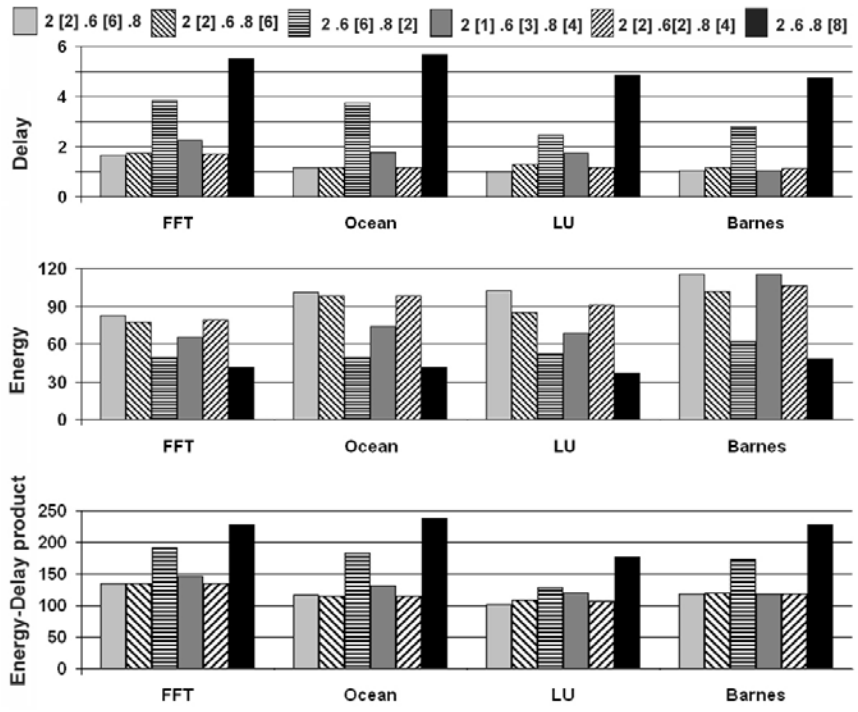


Fig. 2. The 2.6.8 configuration with 8 simultaneous threads

Fig. 2 shows the simulation results for the workload that consists of 8 threads. This figure is composed of three parts that represent power consumption, delay and energy-delay product. The threads that run in each group of cores are shown in brackets. For example (2 . 6 [6] . 8 [2]) configuration means that our workload has eight threads and six of them run on six EV5 and other two threads run on two EV4 cores.

In this experiment we assume 8 threads are available to be run on cores. A smart scheduler that uses the best cores to execute the workload can save both power and time. For example, consider two configurations. The first one is 2 [2] .6 [6] .8 and the second is 2 .6 [6] .8 [2]. The first configuration's energy-delay is 29.5% better than second configuration, but its power consumption is two times worse. Also in second configuration, we can keep high performance cores (two EV6 cores) for serial jobs.

In another example we compare the first configuration of previous example with 2 [2] .6 [2] .8 [4]. Both of them have nearly the same delay, power and energy-delay product. Note that in the first configuration eight EV4 cores remain idle in processor that has the worst delay, but in the second configuration we still have four EV5 and four EV4 cores, that have much better performance than eight EV4. It is concluded that for this situation 2 [1]. 6 [3]. 8[4] configuration is the best, because half of the number of each core are used, and other half remain idle, and scheduler has a better choice for other loads.

Workloads that have high parallelism give more choice to scheduler to assign threads to different cores, and scheduler can choose the best configuration to reach to the goal of the design.

4 Conclusions

In this work we considered a bounded chip area and implemented a multi-core processor whit different set of cores. We used three cores with different levels of performance. Our results showed that a processor that consists of all three cores gives the best delay and energy-delay result. We also found that if energy-delay is the measure of performance, the best collection of cores to execute threads is a mixture of all kinds of cores.

Other design parameters that affect performance of CMPs are L2 cache configuration that is shared between cores, interconnection network, and a smart scheduler that can explore different phases of application and predicts the demand of next phase in order to choose the best set of cores to execute the workload.

References

1. Kumar, R.: Holistic Design for Multi-core Architectures. PhD thesis, university of California, san diego (2006)
2. Huh, J., Burger, D., Keckler, S.: Exploring the design space of future cmps. In: PACT 2001: Proceedings of the 10th International Conference on Parallel Architectures and Compilation Techniques, pp. 199–210, Washington, DC, USA (2001)
3. Kumar, R., Farkas, K., Jouppi, N.P., Ranganathan, P., Tullsen, D.M.: Single-ISA Heterogeneous Multi-Core Architectures:The Potential for Processor Power Reduction. In: Proceedings of the 36th International Symposium on Microarchitecture (Decemeber 2003)

4. Jouppi, N.P., Tullsen, D.M., Kumar, R., Ranganathan, P., Farkas, K.I.: Single-ISA Heterogeneous Multi-Core Architectures for Multithreaded Workload Performance. In: Proceedings of the 31st International Symposium on Computer Architecture (2004)
5. Monchiero, M., Canal, R., González, A.: Design Space Exploration for Multicore Architectures: A Power/Performance/Thermal View. In: ICS 2006 (2005)
6. Woo, S.C., Ohara, M., Torrie, E., Singh, J.P., Gupta, A.: The SPLASH-2 Programs: Characterization and Methodological Considerations. In: Proceedings of the 22nd International Symposium on Computer Architecture, Santa Margherita Ligure, Italy, pp. 24–36. ACM Press, New York (1995)
7. Brooks, D., Martonosi, M.: Dynamic Thermal Management for High-Performance Microprocessors. In: Proceedings of the 7th International Symposium on High-Performance Computer Architecture, Monterrey, Mexico (January 2001)
8. Brooks, D., et al.: Power-aware Microarchitecture: Design and Modeling Challenges for the next-generation microprocessors. IEEE Micro 20(6), 26–44 (2000)
9. Flynn, M.J., Hung, P., Rudd, K.: Deep-Submicron Microprocessor Design Issues. IEEE Micro 19(4), 11–22 (1999)
10. Balakrishnan, S., Rajwar, R.: The Impact of Performance Asymmetry in Emerging Multicore Architectures. In: Proceedings of the 32nd International Symposium on Computer Architecture ISCA 2005 (2005)
11. Moore, G.: Cramming more components onto integrated circuits 38 (1965)
12. Alpha 21064 and Alpha 21064A: Hardware reference Manual. Digital Equipment Corporation (1992)
13. Alpha 21164 Microprocessor: Hardware Reference Manual. Digital Equipment Corporation (1998)
14. Alpha 21264/EV6 Microprocessor: Hardware Reference Manual. Compaq Corporation (1998)
15. Renau, J., Fraguela, B., Tuck, J., Liu, W., Prvulovic, M., Ceze, L., Sarangi, S., Sack, P., Strauss, K., Montesinos, P.: SESC simulator (January 2005), `http://sesc.sourceforge.net`

Reducing the Computational Complexity of an RLS-Based Adaptive Controller in ANVC Applications

Allahyar Montazeri and Javad Poshtan

Electrical Engineering Department, Iran University of Science and Technology,
Tehran, Iran, Narmak 16846
{amontazeri,jposhtan}@iust.ac.ir

Abstract. In this paper a fast array adaptive IIR filter in active noise and vibration control setup is presented. This fast array implementation is an extended form of the fast array algorithms for FIR filter which is studied in literature before. Since the original algorithm derived for ANVC applications was based on RLS recursion its computational complexity was of order $O(n^2)$ and it was also vulnerable to round-off and finite precision errors that may occur in real-time implementation of the algorithm. The proposed fast array solution of this algorithm not only reduces its computational complexity to the order of $O(n)$ with the same performance, but also because of its matrix nature it has good numerical stability in real-time applications which is a necessity in active noise and vibration control applications.

Keywords: ANVC, Fast-array algorithm, IIR RLS-based adaptive filter.

1 Introduction

Currently noise and vibration are known as two main sources of environmental pollution in the entire world. Active noise and vibration control (ANVC) are proved to be used effectively when the incident disturbance are the result of low frequency noise and vibration. In a general view point, active control is defined as a technique for suppressing unwanted disturbances by the introduction of controlled secondary sources such that their outputs interfere destructively with the incident primary disturbance [1].

One of the main problems commonly complained in ANVC applications is slow convergence rate of the algorithms used to adapt the system. In most of cases there is a trade off between the convergence rate of the algorithm and its computational complexity for real-time implementations. One of the fast convergent algorithms reported in ANVC applications uses RLS algorithm with FIR filter structure [2], [3], [4]. This is due to the fact that in contrary to stochastic gradient descent algorithm the convergence behavior of RLS-type algorithms is quite independent of the statistics of the incident noise or vibration signal. The computational complexity of RLS-type algorithms are of the order $O(n^2)$, where n is the length of the control filter. One of the main problems in using plain RLS-type algorithms in ANVC applications is that they suffer from numerical instability due to finite precision computations. To overcome

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 259–266, 2008.

this difficulty it is shown that array-based methods RLS filtering (such as QR algorithm, inverse QR algorithm …) will be more reliable in finite precision implementations [5]. In order to reduce the computational complexity of RLS-type algorithms used in ANVC applications to the order of $O(n)$ floating point operations (flops) per sampling instant fast transversal filter (FTF) is proposed in [4] and its numerical stability is improved by using QR decompositions and lattice structures [3]. However, the comparison study of [6] shows that the performance of FTF implementation of RLS algorithms used in ANVC applications is reduced in comparison with the original RLS algorithm. Array methods [5] are powerful algorithmic variants that are theoretically equivalent to the recursive least-squares algorithms but they nevertheless perform the required computational in a more reliable manner. By exploiting the structure of data in the regression vector, a fast array implementation of RLS algorithm for adapting the FIR filter is derived by [5].

The use of IIR filters in ANVC applications date back to the development of FuLMS algorithm in [7] and after that some improvements is proposed in literatures [8], [9]. By using the *hyperstability* theory, an RLS-type adaptive IIR filter is proposed by the authors of the paper in [10], in which it is shown the performance of the proposed algorithm is superior to the commonly used FuLMS and SHARF algorithms in ANVC applications. Since the computational complexity of the algorithm proposed in [10] is of order $O(n^2)$ and it may be vulnerable to the finite precision implementations and round-off errors, in this paper the fast RLS array implementation of the algorithm is proposed. This array algorithm is developed for IIR filters and is an extension of the algorithms proposed in literatures for FIR filters. It will be shown that the array form of the algorithm exhibits exactly the same performance as the original algorithm while its computational complexity is reduced to the order of $O(n)$. In section 2, the algorithm proposed for adaptation of IIR filter will be briefly introduced, and in section 3 the fast array form of the algorithm is derived. The simulation of the algorithm using identified model of an experimental duct is presented in section 4, and finally some conclusions will wrap up the paper.

2 Efficient RLS-Based Adaptive Controller

The algorithm proposed in [10] is summarized in Table 1, and not repeated here because the lack of space. Considering the update equation in step 7 of Table 1, it is required to compute the gain vector $\mathbf{g}(n)$ to update $\hat{\boldsymbol{\theta}}(n)$. In turn, the evaluation of $\mathbf{g}(n)$ requires the matrix $\mathbf{F}(n)$, and updating $\mathbf{F}(n)$ needs $O(n^2)$ operations per iteration. Besides, the computation of the denominator of $\mathbf{g}(n)$ also requires $O(n^2)$ operation. Since these update steps are the main computational bottleneck in the algorithm, the effort is to develop a time-update for $\mathbf{g}(n)$ directly based on $\mathbf{g}(n-1)$. In order to implement a fast array form of the algorithm of Table 1, the shift structure of data in the regression vector $\boldsymbol{\varphi}_f(n)$ is of great importance. By splitting the regression vector into samples of the previous outputs of the filter, and the samples of the reference signal:

$$\boldsymbol{\varphi}_{1f}(n)=\left[-u'_f(n-1),\cdots\cdots,-u'_f(n-n_A)\right]^T,\ \boldsymbol{\varphi}_{2f}(n)=\left[r_f(n),\cdots\cdots,r_f(n-n_B)\right]^T \tag{1}$$

By defining:

$$\gamma(n+1)=\frac{\lambda_1^{-1}}{1+\frac{\lambda_2}{\lambda_1}\boldsymbol{\varphi}_f^T(n+1)\mathbf{F}(n)\boldsymbol{\varphi}_f(n+1)},\quad \gamma'(n+1)=\lambda_1\,\gamma(n+1) \tag{2}$$

$$\mathbf{g}(n+1)=\frac{\lambda_1^{-1}\mathbf{F}(n)\boldsymbol{\varphi}_f(n+1)}{1+\frac{\lambda_2}{\lambda_1}\boldsymbol{\varphi}_f^T(n+1)\mathbf{F}(n)\boldsymbol{\varphi}_f(n+1)},\quad \mathbf{g}'(n+1)=\frac{\lambda_2}{\lambda_1}\mathbf{g}(n+1) \tag{3}$$

and partitioning the adaptation gain matrix $\mathbf{F}(n)$ as follows:

$$\mathbf{F}(n)=\begin{bmatrix}\mathbf{F}_{11}(n)_{n_A\times n_A} & \mathbf{F}_{12}(n)_{n_A\times(n_B+1)}\\ \mathbf{F}_{21}(n)_{(n_B+1)\times n_A} & \mathbf{F}_{22}(n)_{(n_B+1)\times(n_B+1)}\end{bmatrix} \tag{4}$$

Table 1. Summary of the algorithm proposed in [14]

Steps	Computations
1.Updating regression vector by filtering the new samples	$\boldsymbol{\varphi}_f(n)=\left[-u'_f(n-1),\ldots,-u'_f(n-n_A),r_f(n),\ldots,r_f(n-n_B)\right]^T$ $r_f(k)=\hat{G}_{yu}(q)r(k),u'_f(k)=C(q,k-1)r_f(k)$
2. Calculating the update gain vector	$g(n+1)=\dfrac{F(n)\boldsymbol{\varphi}_f(n+1)}{\lambda_1(n)+\lambda_2(n)\boldsymbol{\varphi}_f^T(n+1)F(n)\boldsymbol{\varphi}_f(n+1)}$
3. Updating covariance of parameter estimation error	$F(n+1)=\dfrac{1}{\lambda_1(n)}\left[F(n)-\dfrac{F(n)\boldsymbol{\varphi}_f(n+1)\boldsymbol{\varphi}_f^T(n+1)F(n)}{\frac{\lambda_1(n)}{\lambda_2(n)}+\boldsymbol{\varphi}_f^T(n+1)F(n)\boldsymbol{\varphi}_f(n+1)}\right]$
4. Measuring *a priori* error signal at error microphone	$e(n+1)=d'(n+1)+G_{yu}(q)\left[\boldsymbol{\varphi}^T(n+1)\hat{\boldsymbol{\theta}}(n)\right]$
5. Calculating filtered *a priori* error	$v_0(n+1)=e(n+1)+\sum_{j=1}^{n_H}h_j\varepsilon(n+1-j)$
6. Calculating *a poste-riori* error	$\varepsilon(n-j)=d'(n-j)+\boldsymbol{\varphi}_f^T(n-j)\hat{\boldsymbol{\theta}}(n-j)$
7. Updating weights of IIR filter	$\hat{\boldsymbol{\theta}}(n+1)=\hat{\boldsymbol{\theta}}(n)-g(n+1)v^0(n+1)$

It can be shown that the time-update equation for the left-hand side of steps 2 and 3 in Table 1 will be rewritten as follows:

$$\begin{bmatrix}\mathbf{g}'_1(n+1)\gamma'^{-1}(n+1)\\ 0\\ \mathbf{g}'_2(n+1)\gamma'^{-1}(n+1)\\ 0\end{bmatrix}=\begin{bmatrix}0\\ \mathbf{g}'_1(n)\gamma'^{-1}(n)\\ 0\\ \mathbf{g}'_2(n)\gamma'^{-1}(n)\end{bmatrix}+\lambda_1^{-1}\delta\mathbf{F}(n)\begin{bmatrix}-u'_f(n)\\ \boldsymbol{\varphi}_{1f}(n)\\ r_f(n+1)\\ \boldsymbol{\varphi}_{2f}(n)\end{bmatrix} \tag{5}$$

$$\gamma'^{-1}(n+1)=\gamma'^{-1}(n)+\begin{bmatrix}-u'_f(n)\\ \boldsymbol{\varphi}_{1f}(n)\\ r_f(n+1)\\ \boldsymbol{\varphi}_{2f}(n)\end{bmatrix}^T\delta\mathbf{F}(n)\begin{bmatrix}-u'_f(n)\\ \boldsymbol{\varphi}_{1f}(n)\\ r_f(n+1)\\ \boldsymbol{\varphi}_{2f}(n)\end{bmatrix} \tag{6}$$

where $\mathbf{g}_1(n)$ and $\mathbf{g}_2(n)$ are obtained by partitioning $\mathbf{g}(n)$ in accordance with the dimensions of the partitioned matrices of $\mathbf{F}(n)$, and $\delta\mathbf{F}(n)$ is calculated as follows:

$$\delta\mathbf{F}(n)=\frac{\lambda_2}{\lambda_1}\left(\begin{bmatrix}\mathbf{F}_{11}(n) & 0 & \mathbf{F}_{12}(n) & 0\\ 0 & 0 & 0 & 0\\ \mathbf{F}_{21}(n) & 0 & \mathbf{F}_{22}(n) & 0\\ 0 & 0 & 0 & 0\end{bmatrix}-\begin{bmatrix}0 & 0 & 0 & 0\\ 0 & \mathbf{F}_{11}(n-1) & 0 & \mathbf{F}_{12}(n-1)\\ 0 & 0 & 0 & 0\\ 0 & \mathbf{F}_{21}(n-1) & 0 & \mathbf{F}_{22}(n-1)\end{bmatrix}\right).$$

Equations (24) and (25) show that the update of the gain vector $\mathbf{g}'(n)\gamma'^{-1}(n)$ as well as $\gamma'^{-1}(n)$ depends only to the value of $\delta\mathbf{F}(n)$, and hence for fast implementation of the algorithm it is required to compute the time-update of $\delta\mathbf{F}(n)$ with the order of $O(n)$ operations. For this purpose the algorithm is initialized by the following parameters:

$$\begin{gathered}\mathbf{g}'(0)=\mathbf{g}(0)=0,\gamma(0)=\lambda_1^{-1},\gamma'(0)=1,\mathbf{F}(0)=\Pi^{-1}=\begin{bmatrix}\Pi_{11}^{-1} & \Pi_{12}^{-1}\\ \Pi_{21}^{-1} & \Pi_{22}^{-1}\end{bmatrix},\mathbf{F}(-1)=\lambda_1\mathbf{F}(0)\\ \Pi_{11}^{-1}=\eta\begin{bmatrix}\lambda_1^2 & \cdots & 0\\ \vdots & \ddots & 0\\ 0 & 0 & \lambda_1^{n_A+1}\end{bmatrix},\Pi_{22}^{-1}=\eta\begin{bmatrix}\lambda_1^{n_A+2} & \cdots & 0\\ \vdots & \ddots & 0\\ 0 & 0 & \lambda_1^{n_A+n_B+2}\end{bmatrix},\Pi_{12}^{-1}=\Pi_{21}^{-1}=0\end{gathered} \tag{7}$$

In this case it can be shown that $\delta\mathbf{F}(0)$ will be initialized with an *M+2* by *M+2* ($M=n_A+n_B+1$) dimension matrix of rank four which has two positive eigenvalues and two negative eigenvalues, and hence can be factored as shown in (8). In (8), $\overline{\mathbf{L}}_0$ is a matrix with the size of $(M+2)\times 4$ and $\mathbf{S}_0$ is the signature matrix. It can be proved that if $\delta\mathbf{F}(n)$ is factorized as $\delta\mathbf{F}(n)=\lambda_1\overline{\mathbf{L}}_n\mathbf{S}_n\overline{\mathbf{L}}_n^T$ ($\mathbf{S}_n,\overline{\mathbf{L}}_n$ has some known property like $\mathbf{S}_0,\overline{\mathbf{L}}_0$), and $\mathbf{F}(n)$ is updated according to step 3 of Table1, then $\delta\mathbf{F}(n+1)$ can also be factored as $\delta\mathbf{F}(n+1)=\lambda_1\overline{\mathbf{L}}_{n+1}\mathbf{S}_{n+1}\overline{\mathbf{L}}_{n+1}^T$ with $\mathbf{S}_{n+1}=\mathbf{S}_n$.

$$\delta\mathbf{F}(0)=\overline{\mathbf{L}}_0\mathbf{S}_0\overline{\mathbf{L}}_0^T,\overline{\mathbf{L}}_0=\sqrt{\eta\lambda_1\lambda_2}\begin{bmatrix}1 & 0 & 0 & 0\\ \vdots & \vdots & \vdots & \vdots\\ 0 & \lambda_1^{\frac{n_A}{2}} & 0 & 0\\ 0 & 0 & \lambda_1^{\frac{n_A}{2}} & 0\\ \vdots & \vdots & \vdots & \vdots\\ 0 & 0 & 0 & \lambda_1^{\frac{M}{2}}\end{bmatrix},\mathbf{S}_0=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & -1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & -1\end{bmatrix} \tag{8}$$

Table 2. Summary of the proposed RLS-Based fast-array adaptive IIR filter

Steps	Computations
1. Initialization of the algorithm by defining parameters λ_1, λ_2 and η.	$\overline{L}_0 = \sqrt{\eta\lambda_1\lambda_2}\begin{bmatrix} 1 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \lambda_1^{\frac{n_A}{2}} & 0 & 0 \\ 0 & 0 & \lambda_1^{\frac{n_A}{2}} & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \lambda_1^{\frac{M}{2}} \end{bmatrix}, S = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}$
2. Updating regression vector in pre-array by filtering the new data.	$r_f(n+1) = \hat{G}_{yu}(q)r(n+1),\ u'_f(n) = C(q,n)r_f(n)$
3. Find a J-unitary transformation matrix Θ_n such that the first element of the first row of the post-array matrix will be positive and its last four elements are equal to zero.	$\begin{bmatrix} \gamma'^{-\frac{1}{2}}(n) & \left[-u'_f(n)\,\boldsymbol{\varphi}_{1f}^T(n)\ r_f(n+1)\,\boldsymbol{\varphi}_{2f}^T(n)\right]\overline{L}_n \\ \begin{bmatrix} 0 \\ g'_1(n)\gamma'^{-\frac{1}{2}}(n) \\ 0 \\ g'_2(n)\gamma'^{-\frac{1}{2}}(n) \end{bmatrix} & \lambda_1^{-1}\overline{L}_n \end{bmatrix}\Theta_n =$
	$\begin{bmatrix} \gamma'^{-\frac{1}{2}}(n+1) & [0\ \ 0\ \ 0\ \ 0] \\ \begin{bmatrix} g'_1(n+1)\gamma'^{-\frac{1}{2}}(n+1) \\ 0 \\ g'_2(n+1)\gamma'^{-\frac{1}{2}}(n+1) \\ 0 \end{bmatrix} & \sqrt{\lambda_1^{-1}}\overline{L}_{n+1} \end{bmatrix}$
4. Extract the required elements of the first column of post-array matrix	$g'(n+1) = \begin{bmatrix} g'_1(n+1)\gamma'^{-\frac{1}{2}}(n+1) \\ g'_2(n+1)\gamma'^{-\frac{1}{2}}(n+1) \end{bmatrix}(\gamma'^{\frac{1}{2}}(n+1))$
	$\gamma'(n+1) = (\gamma'^{\frac{1}{2}}(n+1))(\gamma'^{\frac{1}{2}}(n+1))$
	$g(n+1) = \frac{\lambda_1}{\lambda_2}g'(n+1)\ ,\ \gamma(n+1) = \frac{1}{\lambda_1}\gamma'(n+1)$
5. Measuring *a priori* error signal at error microphone	$e(n+1) = d'(n+1) + G_{yu}(q)\left[\boldsymbol{\varphi}^T(n+1)\hat{\boldsymbol{\theta}}(n)\right]$
6. Calculating filtered *a priori* error	$v_0(n+1) = e(n+1) + \sum_{j=1}^{n_H} h_j\varepsilon(n+1-j)$
7. Calculating *a posteriori* error	$\varepsilon(n+1) = (1 - \frac{\gamma^{-1}(n+1) - \lambda_1}{\lambda_2}.\gamma(n+1))\,e(n+1)$
8. Updating weights of IIR filter	$\hat{\boldsymbol{\theta}}(n+1) = \hat{\boldsymbol{\theta}}(n) - g(n+1)v^0(n+1)$

By using the initialization (7) and (8), and the statement mentioned above, $\delta\mathbf{F}(n)$ in (5) and (6) will be replaced by its factored form. A close inspection of the above equations reveals that they can be written in the following norm preserving and inner-product matrix form [3]:

$$\begin{bmatrix} \mathbf{C} & 0 \\ \mathbf{F} & \mathbf{Z} \end{bmatrix}\begin{bmatrix} \mathbf{C}^T & \mathbf{F}^T \\ 0 & \mathbf{Z} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{D} & \mathbf{E} \end{bmatrix}\begin{bmatrix} \mathbf{I} & 0 \\ 0 & \mathbf{S} \end{bmatrix}\begin{bmatrix} \mathbf{A}^T & \mathbf{D}^T \\ \mathbf{B}^T & \mathbf{E}^T \end{bmatrix} \tag{9}$$

where:

$$\mathbf{A} = \gamma'^{-\frac{1}{2}}(n),\ \mathbf{C} = \gamma'^{-\frac{1}{2}}(n+1),\ \mathbf{S} = \mathbf{S}_n,\ \mathbf{E} = \overline{\mathbf{L}}_n,\ \mathbf{B} = \begin{bmatrix} -u'_f(n) \\ \boldsymbol{\varphi}_{1f}(n) \\ r_f(n+1) \\ \boldsymbol{\varphi}_{2f}(n) \end{bmatrix}^T \overline{\mathbf{L}}_n$$

$$\mathbf{D} = \begin{bmatrix} 0 \\ \mathbf{g}'_1(n)\gamma'^{-\frac{1}{2}}(n) \\ 0 \\ \mathbf{g}'_2(n)\gamma'^{-\frac{1}{2}}(n) \end{bmatrix},\ \mathbf{F} = \begin{bmatrix} \mathbf{g}'_1(n+1)\gamma'^{-\frac{1}{2}}(n+1) \\ 0 \\ \mathbf{g}'_2(n+1)\gamma'^{-\frac{1}{2}}(n+1) \\ 0 \end{bmatrix}.$$

Therefore, there is a $\mathbf{J} = \mathbf{I} \oplus \mathbf{S}$ unitary transformation Θ such that:

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{D} & \mathbf{E} \end{bmatrix}\Theta = \begin{bmatrix} \mathbf{C} & \mathbf{0} \\ \mathbf{F} & \mathbf{Z} \end{bmatrix} \tag{10}$$

and $\Theta\begin{bmatrix} \mathbf{I} & 0 \\ 0 & \mathbf{S} \end{bmatrix}\Theta^T = \begin{bmatrix} \mathbf{I} & 0 \\ 0 & \mathbf{S} \end{bmatrix}$. It can be proved that in (9) $\mathbf{Z} = \sqrt{\lambda_1^{-1}}\overline{\mathbf{L}}_{n+1}$, and hence the array form (10) can be used to update $\mathbf{g}'(n)\gamma'^{-1}(n)$, $\gamma'^{-1}(n)$, and $\overline{\mathbf{L}}_n$.Considering the compact matrix form (10) for updating the weights of IIR filter, the array form of the algorithm listed in Table 1 can be summarized in Table 2. The computational complexity of both algorithms (number of multiplications in each flops) are calculated and compared in Table 3. It is assumed that the secondary path is modeled with an FIR filter of order n_s, and M is the number of coefficients of IIR control filter.

Table 3. Comparison of the computational complexity of both algorithms

Steps	Table 1	Table2		Table 1	Table2
Step 1	n_s+M	-	Step 5	n_H	M
Step 2	$2M^2+M+2$	n_s+M	Step 6	M	n_H
Step 3	$6M^2+M+1$	$25(M+3)$	Step 7	M	4
Step 4	M	$2(M+3)$	Step 8	-	M
Total	Table 1: $8M^2+6M+n_s+n_H+3$		Table 2: $30M+n_s+n_H+85$		

3 Simulation/Experimental Results

The performance of the proposed fast-array adaptive IIR algorithm, is evaluated in an experimental duct shown in Fig. 2. The proposed algorithm is implemented in MATLAB/Simulink environment, and Real-Time Windows Target Toolbox is used to generate C++ codes requires to run the algorithm in real-time manner. The order of

the numerator and denominator of the IIR filter are selected 18 and 11, obtained by identification of primary and secondary path using subspace method. In order to stabilize the algorithm, $H(q)$ is found by trial and error so that the SPR condition requires for the stability of the algorithm satisfied. The algorithm is started with zeros initial conditions for the weights, $\lambda_1 = 1$, $\eta = 10^8$ and λ_2 a very small number. The primary noise is chosen to be white noise and the error signal is measured for 5 second. As can be seen the error signal is converged to zero after about 0.7 second. The convergence behaviour of some of the filter weights are shown in Fig. 4.

Fig. 1. Acoustical experimental duct

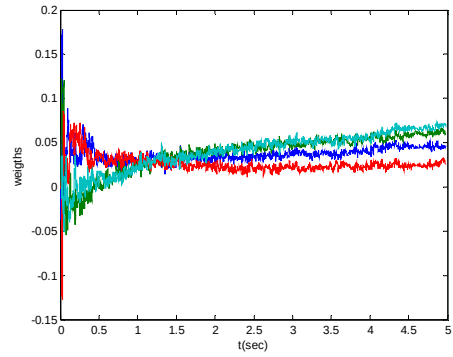


Fig. 2. Convergence behavior of some of the filter weight

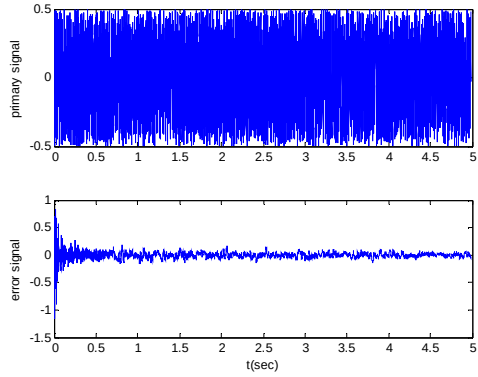


Fig. 3. (up) Primary disturbance signal, (down) error signal at error microphone

4 Conclusions

In this paper an RLS-based adaptive IIR filters in ANVC applications is derived using fast-array method. The proposed algorithm is an extended form of the array algorithm

used for RLS-based adaptive FIR filters in the literature. The computational complexity of the algorithm is of order $O(n)$ in comparison with the original algorithm which is of order $O(n^2)$. Since the RLS-based algorithm suffer from numerical instability, and are vulnerable to round-off and finite precision errors it is necessary to be implemented in a more efficient different form. The performance of the algorithm in the proposed fast-array form will be exactly the same as the original algorithm. The proposed algorithm is tested in an experimental duct using MATLAB/Simulink and real-time Windows Target Toolbox.

References

1. Elliot, S.J.: Signal Processing for Active Control. Academic Press, London (2001)
2. Auspitzer, T., Guicking, D., Elliott, S.J.: Using a Fast-Recursive-Least-Squared Algorithm in a Feedback-Controller. In: IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, pp. 61–64 (1995)
3. Bouchard, M.: Numerically Stable Fast Convergence Least-Squares Algorithms for Multichannel Active Sound Cancellation Systems and Sound Deconvolution Systems. Signal Processing 82(5), 721–736 (2002)
4. Bouchard, M., Quednau, S.: Multichannel Recursive-Least-Squares Algorithms and Fast-Transversal-Filter Algorithms for Active Noise Control and Sound Reproduction Systems. IEEE Transactions on Speech and Audio Processing 8(5), 606–618 (2000)
5. Sayed, A.H.: Fundamentals of Adaptive Filtering. John Wiley & Sons, New Jersey (2003)
6. Diego, M., Gonzalez, A., Ferrer, M., Pinero, G.: An Adaptive Algorithms Comparison for Real Multichannel Active Noise Control. In: Proceeding of 12th European Signal Processing Conference (EUSIPCO 2004), Austria (2004)
7. Eriksson, L.J., Allie, M.C., Greiner, R.A.: The Selection and Application of an IIR Adaptive Filter For Use in Active Sound Attenuation. IEEE Transactions on Acoustics, Speech, and Signal Processing 35(4), 433–437 (1987)
8. Snyder, S.D.: Active control using IIR filters-A second look. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 2, pp. 241–244 (1994)
9. Mosquera, C., Gomez, J.A., Perez, F., Sobreira, M.: Adaptive IIR Filters for Active Noise Control. In: Proceeding of 6th International Congress on Sound and Vibration (1999)
10. Montazeri, A., Poshtan, J.: A Novel Technique for Design and Stability Analysis of Adaptive IIR Filters in ANVC Aplications. In: IEEE International Conference on Signal Processing and Communication (ICSPC 2007), Dubai, U.A.E, vol. 1, pp. 524–527 (2007)

An Efficient and Extendable Modeling Approach for VLIW DSP Processors

Naser Sedaghati-Mokhtari, Mahdi Nazm-Bojnordi, Abbas Hormati, and Sied Mehdi Fakhraie

Silicon Intelligence and VLSI Signal Processing Laboratory
School of ECE, University of Tehran, Tehran 14395-515, Iran
n.sedaghati@ece.ut.ac.ir, m.bojnordi@ece.ut.ac.ir,
abbas@excelicon.com, fakhraie@ut.ac.ir

Abstract. This paper presents an efficient and extendable modeling approach for DSP processors with VLIW architecture. The proposed approach is designed for sequential implementation platforms such as C++ programming language. It employs specific pipeline modeling technique called reverse calling. As a sample implementation, a DSP processor model is designed based on Texas Instruments (TI) C62xx architecture. The processor model handles pipeline resources (memories and register files) during concurrent accesses by updating method. To verify the functionality of the model, a cycle-accurate simulation environment is developed using C++ programming language. In this simulator, a DSP-specific data type, called DSPDT, is designed and implemented for bit-accurate implementation of signal processing operations. The simulation environment utilizes a simple assertion-based verification technique with messages using three levels of severities: Alert, Warning, and Error. The simulator is functionally validated by practical DSP benchmarks such as IIR filters, correlation, FFT blocks and also the G.729a speech codec for single and multiple speech channels.

Keywords: DSP Processors, VLIW Architecture, Cycle-Accurate Simulator.

1 Introduction

Simulation tools are essential aides to both designers and researchers in computer architecture, due to their ability to study and validate new designs without the cost of actually building the hardware. They also provide for a good platform on which researchers can explore a wide range of design choices, which might be practically not feasible. Simulation tools also allow one to study the combined interaction of all the architectural features, before anything is built and can bring out potential bugs that might not have otherwise been detected [1], [2].

In this paper, we present an efficient and extendable processor model for VLIW DSP architectures. We verify the modeling approach by Texas Instruments (TI) C62xx DSP processor. The processor model employs a specific pipeline modeling technique called reverse calling. We present a cycle-accurate simulation environment for the processor model. We validate the simulator by standard DSP benchmarks such as IIR filters, correlation, FFT blocks and G.729a speech codec.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 267–274, 2008.

The rest of the paper will discuss related work in this area of research, modeling issues and approaches, a brief overview on the target processor architecture, environmental considerations, verification and validation strategies, experiments and concluding words.

2 Modeling Approach

Multi-issue VLIW processors usually followed the regular pipeline structure. In contrast with the Superscalar architectures, the pipeline of the VLIW processors is usually free of any conflicting backward paths (i.e., forwarding unit) [3[. Using this benefit, we propose a modeling technique for VLIW DSP architectures (including pipeline structure, register file and memory, and processor core) in sequential platforms (i.e. C++ programming language).

Before describing the modeling approach, we present two concepts in this area.

2.1 Single vs. Multiple Stage Units

According to the timing and access methodology, the processor pipeline has two kinds of units: single-stage and multi-stage. Majority of the units in a VLIW processor model are single-stage in which the overall function of that unit is started, continued, and terminated in the same pipeline stage. Example of these units is decoding stage(s). In each cycle, each single-stage unit receives requests only from preceding neighbored unit in the processor pipeline. In other words, the input and output interfaces of a unit in stage N are with stages N-1 and N+1, respectively. Another group of units, multi-stage units, are that influenced by the pipeline in more than one stages in each execution cycle. Generally, the entire pipeline shared resources such as register file, memory and I/O interfaces are multi-stage units.

2.2 Single vs. Multiple Cycle Units

For stage-based categorization, we consider simulation cycle which is demonstrated spatial distribution of VLIW instruction packets. Changing the point of view from spatial to temporal distribution of instruction packets, the processor units divided into two categories: Single-cycle and multi-cycle. Dispatch is the only multi-cycle unit which is continued operating for a single instruction packet to one cycle or more. Other simulator units belong to single-cycle category.

Accordingly, multi-cycle objects produce new handling issues such as pipeline exceptions and interfaces. On the other hand, multi-stage objects require special considerations for concurrent accesses. These issues are addressed by message passing strategy and updating mechanism.

2.3 Message Passing Strategy

Implementing the precise pipeline structure imposes a message passing strategy. In this simplest form, all of the output variables of each pipeline stage objects are considered as message variables. The pipeline stage objects communicate together through these variables. Each object has a *run()* method which is called when the

corresponding operation is required. For maintaining the pipeline structure, we call the *run()* method of all pipeline objects in the reverse order. Thus, we called this strategy as Reverse Calling. This technique is used to prevent undesirable transmissions of message variables through pipeline stages.

2.4 Updating Mechanism

To handle multiple access requests, each of the shared resources (multi-stage objects) has a method called *update()*. At the execution time, some components send their requests to the target multi-stage objects. At the end of simulation cycle, and when all requests are received and gathered, the target object updates its contents. This way, validations and access limitations are checked during the updating step.

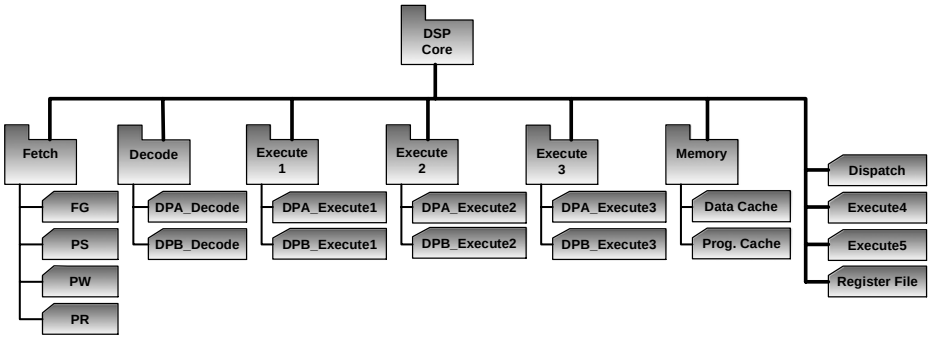


Fig. 1. Class Hierarchy of the processor model

3 Processor Model

We exemplify the modeling approach by Texas Instruments (TI) C62xx DSP processor [4], [5]. It is a fixed-point device employed the VelociTI™ architecture. The core of the CPU consists of 32 general purpose registers of 32-bit lengths, which can be combined to store 16 long values of 40-bit lengths. The core can execute up to eight instructions every cycle, one each for the eight functional units it has. The functional units are divided into two similar sets of basic functional units. Each set can access its own one half of the register file directly and the other half can be accessed using cross paths. The two halves are named 'A' and 'B'. More architecture details are available at [4], [5].

The exemplified model consists of several numbers of classes which are familiar to the hardware modeling concepts. The hierarchical design for classes makes the implementation process easier and less complicated. Each block of a pipeline stage is considered as an independent class with its own methods and attributes. The main class hierarchy is demonstrated in Fig 1.

3.1 Pipeline Model

For modeling the real pipelining operation, we employed the clocking concept of the real processor hardware. At the clock edge, each stage component is fired. This

caused the run method of each component to be called at the time of activation. The delay elements (registers) have not been modeled explicitly in the simulator; they are considered implicitly embedded by reverse calling and updating mechanisms. The update mechanism for shared resources is performed at the end of each simulation cycle.

Connecting all the stages from Fetch to Execute in a forward direction may cause propagation of unexpected faulty values into the pipeline stages. Avoiding this problem, all the pipeline stages are executed in the reverse direction, called reverse calling technique. Using this strategy, at the beginning of each simulation cycle, each stage will have its registered values from the previous stage. These values are actually updated at the previous simulation cycle (by calling the *run()* methods of pipeline objects) and correctly utilized in the current cycle.

3.2 Register File Model

General register file model consists of two internal parts (A and B) which are instantiated from the same class, i.e. *RegFilePart*. Considering real reading and writing constraints, a controlling mechanism is employed to optimize port assignment to the functional units and prevent any unpredictable conditions. Each register file part (A and B) has 10 read ports and 6 write ports. One of the read ports is dedicated to cross read operations. For practical implementation, and according to the decoding information, a multiplexing mechanism is employed to direct the read registers from the limited read ports to the desired source operands of the requesting functional units.

For the write operation, where the number of writes is greater than the number of available ports, we multiplex the operations and assign them to the available write ports. In this way, when all the functional units perform their write accesses to the register files (add their write request address to the write address queue), and therefore create a write transaction, the requested accesses are processed.

Limited numbers of selected writes will be directed to the available write ports. In this way, after the termination of all executions in a simulation cycle, the register file is updated. This means that all the pending write requests are committed at the end of each simulation cycle. For the case of multiple assignments to the same port, multiplexing with suitable priority mechanisms are considered to make the result of all the register file accesses fully predictable.

It should be noted that, from the assembler point of view, it is impossible for a register file to be accessed (read or written) more than the number of available ports. It is also impossible to be written more than one to a single write port in a single cycle.

3.3 Memory Model

Two kinds of memories are designed for the environment: Program and Data memories. All the memories' parameters are selected according to the TI's reference model. The Fetch stages (PG, PS, PW, and PR) are interfaced to the Program memory while D units are interfaced to the Data memory. Memory models provide all the real memory and caching behaviors. Therefore, one can easily use the proposed environment for analysis and evaluation of the DSP applications according to its reported run-time statistics.

4 Processor Simulator

For implementing the model and verifying its functionality, we develop a cycle-accurate simulation environment. The simulator is intended to simulate the real behavior and function of the target VLIW processor. In order to execute real bit-accurate DSP operations of various bit widths, the simulator employs a specific type of data as a C++ class which is called DSPDT.

4.1 Implementation

At the center of the simulator, there is a specific data type designed and developed for accurate DSP simulation. The data type, called *DSPDT*, is developed to model all the DSP operations practically. The simulator blocks are working with signals which are only inherited from this data type. The *DSPDT* methods and attributes make available an accurate, bit-true data type for modeling and simulation of DSP operations in C++. Also, this data type provides capability to monitor objects of the simulation environment based on real DSP behaviors, such as saturation.

4.2 The Simulator Architecture

The simulator model, in addition to datapath and controller, consists of memory, register file, and statistical reporting and monitoring (SRMU) units, as shown in Fig 2-a.

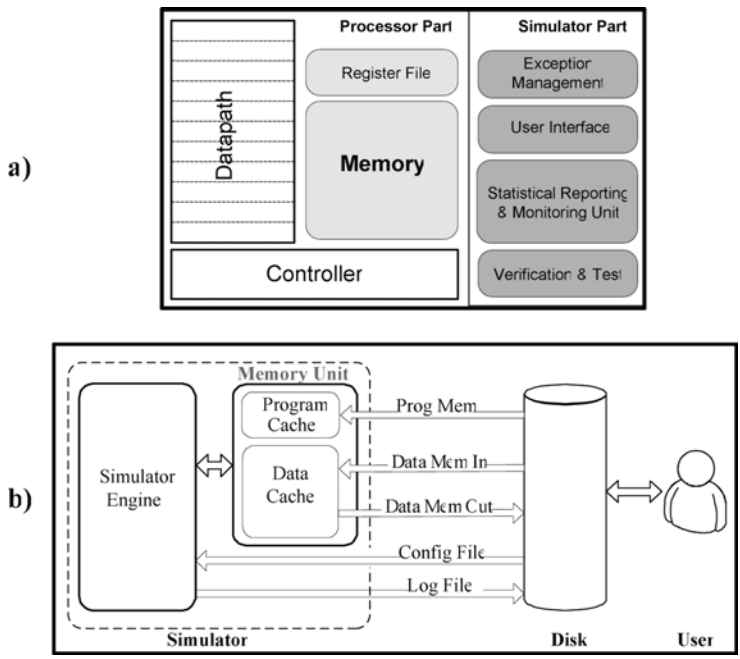


Fig. 2. a) Simulator architecture and b) Interaction between user and simulator

The SRMU is responsible for gathering all the user-requested information from the simulation. This unit is interfaced to standard file system to store the monitoring and statistical reports in the desired files. Employing such a file-based interfacing mechanism, user is able to access and control the simulator easily. Fig 2-b shows the possible relation between user and the simulation environment.

Four consequent operations construct the main simulation process. At the first step, the simulator provides a copy of all the pipeline controlling and state variables. The pipeline objects utilize these copies at each cycle. At the second step, the simulator call the *run()* methods of the stage objects. This step includes synchronization and providing message variables. Updating shared resources through update mechanism is the third step of simulation round. Finally, by calling the SRMU, simulator reports the statistics gathered during the monitoring of the model.

5 Experiments

We have integrated our model using all the classes with their methods and attributes. The developed model is compiled using GNU C++ Compiler. The testing strategy is based on TI's CCStudio toolset. We have applied tests to both models: ours and CCStudio. For validation, DSP microbenchmarks such as IIR filters, Correlation and FFT block, and so on are utilized. Running these microbenchmarks on the proposed simulator is feasible by the cycle count listed in Table 1Table .

For IIR filters, *n* determines number of cycles required for processing of each input sample. For FFT, *n* is the size of the FFT block. In autocorrelation, *n* and *m* determine number and length of correlations, respectively. We note that mentioned delay clock cycles are considered with counting no cache miss cycle. Executions of the mentioned benchmarks are validated by Code Composer Studio toolset [8] and verified by the developed bit-true models in MATLAB.

Table 1. Execution of real DSP microbenchmarks

µbenchmark	Clock Cycle
4th order IIR	$10 \times n + 6$
6th order IIR	$15 \times n + 6$
Radix-4 FFT	$Log_4(n) \times (10 \times n/4 + 29) + 36 + n/4$
Autocorrelation	$n \times m/2 + 31 + (n/2\text{-}1)$

For simulation of the voice processing applications, we employ G.729a 6] standard reference code which is developed and released in C language. We modify the code for real DSP execution. Table 2 demonstrates the obtained results of G.729a speech codec for 10ms (one frame) single-channel voice data. Although the code is not obviously well optimized, but it satisfies the simulation requirements for application evaluation.

Table 2. Execution of single channel G729a

Measure	Instruction Count	Simulation Cycles
Encoder	213526	81565
Decoder	90541	34823

Regarding multi-channel simulation of G.729a speech codec, we modify the reference ITU code to support reentrancy capability. The obtained results demonstrate that, considering the real processor implementation, the model supports up to 10 real-time voice channels while theoretically the architecture should be able to process more real-time voice channels [7]. This limitation is imposed because of the inefficiency of the application code, many frequent memory accesses and context-switching overheads during multichannel operations. The performance and also implementation complexity of the proposed approach, in comparison with the previously presented models (C6XSin [8] and SimpleDSP [10]) approve the achieved performance gain.

6 Conclusion

In this paper, we have presented an efficient and extendable modeling approach for DSP architectures with VLIW architectures. For this purpose, a cycle-accurate simulation environment based on reverse calling technique is designed and presented. The environment is verified by developing processor model of the Texas Instruments TMS320C6211 (C62xx) DSP processor using C++ programming language. A DSP-specific data type, called DSPDT, is designed and utilized for real implementation of practical processor environment. An assertion-based verification is employed to verify the code and provide messaging and monitoring facilities in different levels. The simulator, and accordingly the modeling approach, is functionally validated using DSP benchmarks employing such as IIR filters, autocorrelations, FFT blocks, and G.729a speech codec.

Acknowledgments. Naser Sedaghati-Mokhtari wishes to express his gratitude to Iran Telecommunication Research Center (ITRC) for financial support of his research.

References

1. Rosenblum, M., Herrod, S.A., Witchel, E., Gupta, A.: Complete Computer Simulation: The SimOS Approach. IEEE Parallel and Distributed Technology 3, 34–43 (1995)
2. Bechem, C., Combs, J., Utamaphethai, N., Black, B., Shawn Blanton, R.D., Shen, J.P.: An Integrated Functional Performance Simulator. IEEE MICRO 19, 26–35 (1999)
3. Patterson, D.A., Hennessy, J.L.: Computer Architecture: A Quantitative Approach, 2nd edn. Morgan Kaufmann, Menlo Park (1996)
4. Turley, J., Hakkarainen, H.: TI's New C6x DSP Screams at 1,600 MIPS. Microprocessor Report 11, 1–4 (1997)
5. TMS320C6000 CPU Instruction Set Reference Guide. Texas Instruments Literature Number SPRU189C (1999)
6. ITU-T Draft Recommendation G.729. Coding of Speech at 8Kbps Using the Conjugate Structure Algebraic Code Excited Linear – Prediction (CSACELP) (1996)
7. TI SPRA564B. G.729/A Speech Coder: Multichannel TMS320C62x Implementation (February 2000), `http://www.ti.com`

8. Texas Instruments. Code Composer Studio IDE Version 2.2 (August. 2003) http://www.ti.com/tmwccs
9. Ringenberg, J., Oehmke, D., Austin, T., Mudge, T.: SimpleDSP: A Fast and Flexible DSP Processor Model. In: Workshop on Media and Streaming Processors (MSP5) in IEEE/ACM MICRO-36, San Diego (2003)
10. Barbieri, I., Bariani, M., Raggio, M.: A VLIW architecture simulator innovative approach for HW-SW co-design. In: IEEE International Conference on Multimedia and Expo (ICME), New York, pp. 1375–1378 (2000)

An Exact Algorithm for the Multiple-Choice Multidimensional Knapsack Based on the Core

Mohammad Reza Razzazi and Taha Ghasemi

CE and IT Department, Amirkabir University of Technology, Tehran, Iran
{razzazi,tghasemi}@aut.ac.ir

Abstract. In this paper, we propose a branch and bound algorithm to solve the multiple-choice multidimensional knapsack problem. The branch and bound tree is arranged based on the orderings in the core and navigated in a depth first manner, which while consuming low memory effectively causes the core to be expanded by need. We use Osorio's mixed constraint and the linear programming solution of the surrogated problem for bounding tests. The computational results and comparison with the previous best exact algorithm shows that the algorithm has significant performance improvement over the previous algorithm.

1 Introduction

The multiple-choice knapsack problem (MCKP) is a generalization of 0-1 knapsack problem (KP). In MCKP we have several classes of items; each item has its own profit and weight. The goal is to choose exactly one item from each class in order to maximize the sum of items profits while satisfying the weight constraint. The multiple-choice multidimensional knapsack (MMKP) generalizes MCKP by considering several constraints instead of just one. Formally, MMKP can be stated as follows:

$$\begin{aligned} &\max \sum_{i\in C}\sum_{j\in N_i} p_{ij}x_{ij} \\ &s.t. \sum_{i\in C}\sum_{j\in N_i} w_{ij}^{k}x_{ij} \le b^{k} \quad k\in\{1,..,m\} \end{aligned} \tag{1}$$

$$\sum_{j\in N_i} x_{ij} = 1 \quad i\in C \tag{2}$$

$$x_{ij}\in\{0,1\} \quad i\in C, j\in N_i \tag{3}$$

where C is the set of classes with cardinality g, N_i is the index set of class i items with cardinality n_i, p_{ij} is the profit of j^{th} item of class i and w^k_{ij} is its weight in k^{th} constraint, and x_{ij}s are decision 0-1 variables which must be identified and determine whether corresponding item is chosen or not. Constraint (2) states that exactly one variable from each class must be equal to one. If we relax integrality constraint (3) and allow x_{ij} to take on fractional values from 0 to 1, the resulted problem is called LMMKP. Let $n = \max\{n_i\}$.

MMKP contains KP as a special case so it is NP-hard. Thus, most of the works in the MMKP area are devoted to heuristic solutions of the problem [1-5]. As our best of

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 275–282, 2008.

knowledge there are only two exact algorithms directly dealing with MMKP. In [1], a branch and bound (B&B) algorithm has been developed to solve MMKP. The B&B tree is navigated in a best first manner and the solution of LMMKP, using the simplex algorithm, is used for upper bound testing. More recently, [6] has proposed an algorithm, which is superior to the previous one. The idea is to enumerate from the most profitable solution down to the least one until reaching the first feasible solution, which is the answer. The enumeration is done by using best first navigation of the B&B tree. Due to best first nature of both algorithms, they are both memory consuming and hence are not able to solve medium to large instances.

The core concept was first introduced to solve large KP problems [7]. Balas and Zemel observed that there is little difference between the optimal solution of KP and the optimal solution of its linear relaxation (in randomly generated instances). They called the minimum interval containing non-equal variables the core, assuming items are sorted in non-increasing profit to weight ratios. Only approximation of the core can be found before exactly solving the problem. The core can be approximated by predetermining its size. In spite of this, introduction of the core resulted in efficient algorithms for solving KP. B&B algorithms use the approximate core by first obtaining a (near) optimal solution in the core sub space. They then try to prove the solution is optimal. If this process fails, the near optimal solution is used as a good starting lower bound for usual B&B algorithms for KP [8]. Pisinger incorporated the core in a dynamic programming approach for dynamic expansion during the solution process and obtained very efficient algorithms for solving KP [9] and MCKP [10].

Success of the core in the area of knapsack motivates us to apply the core in an exact algorithm for solving MMKP. Up to now, no one has applied the core to exactly solve the multidimensional problems. In multidimensional cases, dynamic programming encounters curse of dimensionality. In addition, the core has not been embedded in B&B approaches properly. In this paper, we first identify an approximate core for MMKP. We then incorporate it, in a new way, to a B&B algorithm by arranging the tree completely based on the core. The tree is navigated in a depth first manner, which while consuming little memory effectively searches around the core sub space. This also can be seen as dynamic expansion of the approximate core by need. The computational experiences show that the resulted algorithm has considerable performance improvement upon the previous best algorithm and is capable of solving larger instances because of its low memory consumption.

The paper is organized as follows. Section 2 gives some background information about the MCKP. In section 3 components of the main algorithm is presented and in section 4 computational experiences and comparison of the algorithms are discussed.

2 Multiple-Choice Knapsack

The multiple-choice knapsack problem (MCKP) is a MMKP with only one constraint. Later we will apply an approach to relax MMKP as MCKP for upper bound testing. In this section some basic properties of MCKP is described. It has been shown that MCKP and its linear relaxation (LMCKP) have following 3 properties [11]. (We use the notation of MMKP for MCKP except the constraint number indexes are dropped.)

1. If $w_{ij} \geq w_{ik}$ and $p_{ij} \leq p_{ik}$ then $x_{ij} = 0$ in every optimal solution of MCKP. In other words, item k dominated item j in class i because it has less weight while give us more profit. Fig. 1 shows a sample class. Each point represents an item in that class. Items that are outside of the shaded region are dominated by at least one item.
2. If $\frac{p_{ij} - p_{ih}}{w_{ij} - w_{ih}} > \frac{p_{ij} - p_{ik}}{w_{ij} - w_{ik}}$ then $x_{ik} = 0$ in every optimal solution of LMCKP. This property implies that in order to solve LMCKP, in each class, we only need to consider items that make lines of the upper convex hull. So for example, only bolded points in Fig. 1 are of interest for LMCKP. Let L_i be the set of such points.
3. In every optimal solution of MCKP, variables of all the classes, except at most one class, are 0 or 1. The exceptional class has two fractional variables, which are the two ends of a line in the upper convex hull.

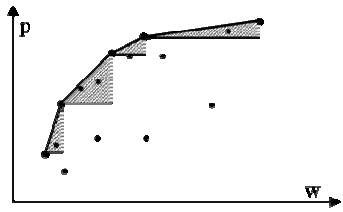


Fig. 1. A sample class in MCKP

Using properties 2 and 3 one can solve LMCKP with the following algorithm [10]:

1. Remove the items in N_i-L_i, and sort the remained items in increasing order of their weights in each class.
2. Select the first (lightest) item from each class and set $p = \sum_{i \in C} p_{i,1}$, $w = \sum_{i \in C} w_{i,1}$, $x_{i,1} = 1 \forall i \in C$.
3. Calculate slope of lines: $\lambda_{ij} = (p_{ij} - p_{i,j-1})/(w_{ij} - w_{i,j-1})$. This value indicates changes of efficiency if we choose item j instead of item j-1 in class i.
4. Sort the lines in non-increasing order of their slopes. Start from the first line.
5. Select the next line with the slope λ_{ij}. If $w + w_{ij} \geq b$ go to step 6. Else update variables and repeat this step. $p = p + p_{ij} - p_{ij-1}$, $w = w + w_{ij} - w_{ij-1}$, $x_{ij} = 1$, $x_{i,j-1} = 0$.
6. If $w = b$ the algorithm is finished. Else assume λ_{ij} was the last line (the break line) slope. Calculate two fractional variables: $x_{ij} = (b - w)/(w_{ij} - w_{i,j-1})$, $x_{ij-1} = 1 - x_{ij}$. And the optimal objective value would be: $p + (b - w)\lambda_{ij}$.

3 The Main Algorithm

In this section, the main components of a B&B algorithm for solving MMKP are presented. The structure of the tree, navigation method, and bounding techniques are discussed.

3.1 Conversion to MCKP

We use the surrogate technique to relax MMKP to MCKP. Resultant MCKP, which we call it the associated MCKP, will be used for upper bound testing and computation of the core. In the surrogate technique, multiple constraints are replaced with a linear combination of them [12]. We approximate the surrogate multipliers by using dual values of the constraints in the LMMKP.

3.2 MMKP Core

We can see the approximate core, in a more general view, as a promising subspace of the solution space which has a near optimal solution. In KP case, this subspace is obtained by fixing variables before the core interval to 1 and after the core interval to 0. This is done after sorting the items according to non-increasing profit to weight ratios. For dynamic expansion of the approximate core, we may use this ordering to increase size of the core interval gradually. In MMKP, we can define two orderings instead of one, one for ordering of the classes and one for ordering of the items in each class. Assume classes are sorted according to the classes ordering and items of classes according to the ordering of items. The approximate core subspace can be obtained by taking first k_1 classes and in these classes taking first k_2 items. Variables of the other $g-k_1$ classes are set to the first item. Variables of the other n_i-k_2 items in class i ($i=1..k_1$) are set to zero.

The orderings for MMKP are experimentally selected as follows. For ordering the classes, for each class we define the break difference as:

$$D_i = \min_{j \in L_i} \{ \left| \lambda^* - \lambda_{ij} \right| \} \quad i \in C \tag{4}$$

where λ^* is the break slope of the associated LMCKP and λ_{ij} and L_i are as defined in section 2 for the associated LMCKP. D_i measures how close a line of a class to the break line therefore if a class has small D_i it will have more potential to be like the break class and to take different value in its optimal solution. Thus, we order classes according to increasing D_i.

We chose to sort items in each class according to increasing of their absolute reduced costs in LMMKP. The reduced cost of a variable is the amount of change in the objective value of a linear programming problem if the value of that variable is changed by 1. Therefore, in each class, items with low absolute reduced cost may be changed with the first item in the class without possibly affecting the objective value too much. Thus, they have more potential to be in the optimal solution. Now we can start from a small (1-class, 1-item) core and expand it by need. This approach is explained in the next subsection.

3.3 Branching Method

Instead of taking a fixed size core, we arrange the B&B tree in such a way that the core will be expanded during its navigation. Suppose we arranged the classes and the items in each class according to the orderings in the MMKP core defined above. The tree is arranged based on these orderings. The tree has g levels. Level i of the tree corresponds to $(g-i+1)^{th}$ class. The branches in level i corresponds to the items of class i. A node at the i^{th} level and j^{th} branch is represented by n_{ij}.

The tree is navigated in a depth first manner. During the navigation when taking n_{ij} x_{ij} is set to 1 and the other variables in this class to 0. Classes that are at the top of the tree are less likely to take different value than their first item. Therefore, by this type of navigation we fix classes that there is more confidence about their values (making good decisions at first) and try to find values of the other classes. This type of navigating also consumes memory proportional to the depth of the tree which is negligible. When we reach to the bottom of the tree, we update the best solution if a better feasible solution is found.

After adding bounding tests of the next subsection, when the algorithm navigates this tree it simulates the following behavior. First begins by a core with the first class and fixes variables of the other classes. The algorithm then completely enumerates the subspace induced by this core. If optimality could not be proved (using bounding tests), it will expand the core with the next most probable changing class and completely enumerate the subspace of this core, and so on. Because the items are also sorted, the algorithm first examines the most probable candidates in each class hence we can expect to reach the optimal solution quickly.

3.4 Bounding Methods

In order to effectively prune the sub trees we test a node for fathoming at three steps. If a node is failed in one of these tests it will be fathomed. For the first test, we use propagation of Osorio's mixed constraint [13]. We observed that this constraint, in addition of its usage for fixation as used by Osorio, can be propagated like the other constraints. Because it is a valid constraint if it is violated in any node we can fathom that node. We used the objective value of the best feasible solution (incumbent) as the lower bound.

In the second test, we check that whether a node leads to a feasible solution or not. This is done by (6). n_{ij} is fathomed if:

$$\exists k \in \{1,..,m\}: \quad \sum_{l=g}^{i} w^{k}_{l,s_l} + \sum_{l=i-1}^{1} \min_{j \in N_l} \{w^{k}_{l,j}\} > b^{k} \tag{5}$$

Equation (6) states that even if one constraint is violated by adding the consumed amount of that constraint so far and the minimum consumption of the remained classes, we can be sure that there is no feasible solution from this node. In each node, we keep sum of the consumed amount of each constraint, hence, by using information of the parent node we can compute the above summation incrementally.

Finally, we compute an upper bound using the optimal solution of the associated LMCKP and compare it to the incumbent objective value. If it is less than or equal to the incumbent objective value we will fathom the node. This step is put last because it

is more time consuming than the two previous steps. We use the algorithm presented in section 2 to solve LMCKP. However to do it faster, at the root we solve the associated LMCKP for once and obtain the break line and store the lines in a list. For other nodes, instead of starting from the first line at each node, we keep track of changes with respect to the root and start from the break line.

4 Computational Results

In order to test the algorithms in practice, all the algorithms were implemented with C++. The algorithms were run over a P4 3.4 GHz computer. We used LP-Solve 5.5 library for solving LMMKP and computing the dual values.

We generate two classes of test problems. In Random or Uncorrelated (UC) class, the profits and weights are drawn from a uniform integer generator in the range of [1,1000]. The right hand sides are computed according to (7).

$$b^k = (\sum_{i \in C} \max_{j \in N_i} \{w_{ij}^k\} + \sum_{i \in C} \min_{j \in N_i} \{w_{ij}^k\})/2 \tag{6}$$

In the second class we generate more correlated instances. Experiences in knapsack problems have shown that instances which have correlation between their profits and weights are more difficult to solve. [14] proposed following equation for multidimensional knapsacks to generate correlated instances by computing profits from weights.

$$p_{ij} = \sum_{k=1}^{m} w_{ij}^k / m + 500.q_{ij} \quad i \in C, j \in N_i, q_{ij} \in U(0,1) \tag{7}$$

where U is a uniform random generator. For the second class (class C) the weights and right hand sides are computed like class UC and the profits are computed using (8). In each class (UC and C) we generate instances according to different configurations. A configuration is represented by a triple (g,n,m). For each configuration, 36 instances were generated and solved and the results were averaged over them.

We compared our algorithm with the EMKP algorithm, which is described in [6]. According to [6], EMKP completely dominates the algorithm developed in [1] and can be consider as the best algorithm for solving the problem. Tables 1 and 2 show results of running both algorithms against problems with different configurations for classes UC and C respectively. Entry with the value of 'space' ('time') indicates that the corresponding algorithm ran out of its memory (time) for at least one instance. The maximum available memory for each algorithm is 512 MB and the maximum time is set to 1 hour. As the results show, the core based algorithm, which is developed here, significantly performs better than EMKP. The core-based algorithm has small standard deviation, which makes it a stable algorithm with respect to different instances. Due to best first nature of EMKP, it cannot solve medium to large instances because of memory shortage. In class UC instances, the core-based algorithm works very fast. Instances of this class have small duality gap (difference between the optimal value and the optimal value of the linear relaxation of the problem). Using the core, the algorithm quickly finds the optimal solution and using the upper bound testings, it prunes large portion of the tree and proves optimality of the solution.

Table 1. The average running time (standard deviation) of algorithms in seconds for different configurations of class UC over 36 instances

Configuration	Core based	EMKP
10,10,2	0.00 (0.01)	0.00 (0.01)
10,10,5	0.01 (0.01)	0.02 (0.09)
10,10,10	0.01 (0.01)	0.09 (0.15)
20,50,2	0.05 (0.02)	0.05 (0.26)
20,50,5	0.09 (0.02)	0.90 (3.99)
20,50,10	0.15 (0.02)	(space)
50,20,2	0.12 (0.02)	(space)
50,20,5	0.17 (0.02)	(space)
50,20,10	0.39 (0.66)	(space)

Table 2. The average running time (standard deviation) of algorithms in seconds for different configurations of class C over 36 instances

Configuration	Core based	EMKP
10,10,2	0.00 (0.01)	21.77 (19.43)
10,10,5	0.01 (0.01)	20.86 (18.62)
10,10,10	0.06 (0.05)	37.74 (35.93)
20,50,2	0.07 (0.01)	(space)
20,50,5	25.67 (27.5)	(space)
20,50,10	(time)	(space)
50,20,2	0.13 (0.01)	(space)
50,20,5	130.59 (150.01)	(space)
50,20,10	(time)	(space)

In class C instances duality gap is high. Thus, EMKP considers too many unfeasible states before reaching the optimal state. Hence, even for small instances, EMKP encounters lack of memory. Highness of duality gap makes fathoming tests and variable fixation less effective. Thus, the core-based algorithm also must spend more time exploring the tree to prove the optimality of the solution.

5 Conclusion

In this paper a branch and bound algorithm for exactly solving the multiple-choice multidimensional knapsack problem (MMKP) is described. The branch and bound tree was arranged according to the defined orderings in the core and was navigated in a depth first manner. This caused the dynamic expansion of the core and low memory consumption. Pruning of sub trees were done at three steps: checking Osorio's mixed constraint, checking feasibility, and checking the upper bound computed with the aid of the associated LMCKP. As the results showed, the algorithm works very well for randomly generated instances but there is need for further investigations to develop more efficient algorithms to solve harder instances of MMKP.

Acknowledgments

The authors would like to thank from 'Iran Telecommunication Research Center' (ITRC) for their support from this project.

References

1. Shahadatullah-Khan, M.: Quality adaptation in a multisession multimedia system: Model, algorithms and architecture, Ph.D. dissertation, Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC, Canada (1998)
2. Parra-Hernandez, R., Dimopoulos, N.J.: A New Heuristic for Solving the Multichoice Multidimensional Knapsack Problem. IEEE Transactions on Systems, Man, and Cybernetics 35(5), 708–717 (2005)
3. Hifi, M., Michrafy, M., Sbihi, A.: A Reactive Local Search-Based Algorithm for the Multiple-Choice Multi- Dimensional Knapsack Problem. Computational Optimization and Applications 33(2-3), 271–285 (2006)
4. Akbar, M.M., Manning, E.G., Shoja, G.C., Khan, S.: Heuristic Solutions for the Multiple-Choice Multi-Dimension Knapsack Problem. In: Alexandrov, V.N., Dongarra, J., Juliano, B.A., Renner, R.S., Tan, C.J.K. (eds.) ICCS-ComputSci 2001. LNCS, vol. 2074, pp. 659–668. Springer, Heidelberg (2001)
5. Mostofa Akbar, M., et al.: Solving Multidimensional Multiple-choice Knapsack by constructing convex hull. Computers and Operations Research 33(5), 1259–1273 (2006)
6. Sbihi, A.: A best first search exact algorithm for the Multiple-choice Multidimensional Knapsack Problem. Journal of Combinatorial Optimization 13(4), 337–351 (2007)
7. Balas, E., Zemel, E.: An algorithm for large zero-one knapsack problems. Operations Research 28, 1130–1154 (1980)
8. Martello, S., Toth, P.: Knapsack Problems: Algorithms and Computer Implementations. J. Wiley, Chichester (1990)
9. Pisinger, D.: A minimal algorithm for the 0-1 knapsack problem. Operations Research 45, 758–767 (1997)
10. Pisinger, D.: A minimal algorithm for the Multiple-choice Knapsack Problem. European Journal of Operational Research 83, 394–410 (1995)
11. Sinha, P., Zoltners, A.: The Multiple-Choice Knapsack Problem. Operations Research 27(3), 503–515 (1979)
12. Glover, F.: Surrogate Constraint Duality in Mathematical Programming. Operations Research 23(3), 434–445 (1975)
13. Freville, A., Plateau, G.: An Efficient Preprocessing Procedure for the Multidimensional 0-1 Knapsack Problem. Discrete Applied Mathematics 49, 189–212 (1994)
14. Osorio, M.A., Hernandez, E.G.: Cutting Analysis for MKP. In: Proceedings of the Fifth Mexican International Conference in Computer Science, pp. 298–303 (2004)

Kinetic Polar Diagram

Mojtaba Nouri Bygi[1,2], Fatemeh Chitforoush[1],
Maryam Yazdandoost[1], and Mohammad Ghodsi[1,2,⋆]

[1] Department of Computer Engineering, Sharif University of Technology,
P.O. Box 11365-9517, Tehran, Iran
[2] School of Computer Science, Institute for Studies in Theoretical Physics and Mathematics,
P.O. Box 19395-5746, Tehran, Iran

Abstract. Polar Diagram [4] is a new locus approach for problems processing angles. The solution to many important problems in Computational Geometry requires some kind of angle processing of the data input. Using the Polar Diagrma as preprocessing, exhaustive searches to find those sites with smallest angle become unnecessary.

In this paper, we use the notion of kinetic data structure [1][2] to model the dynamic case of polar diagram, i.e we maintain the polar diagram of a set of continuously moving objects in the scene. We show that our proposed structure meets the main criteria of a good KDS.

Keywords: Polar diagram, Kinetic data structures, Geometric events, Computational Geometry.

1 Introduction

C. I. Grima et al. [4] introduced the Polar Diagram. The polar diagram of the scene E, consisting of n two-dimensional objects, $E = \{o_0, o_1, \ldots, o_{n-1}\}$, denoted as $\mathcal{P}(E)$, is a plane partition in polar regions. Each generator object o_i creates a polar region $\mathcal{P}_E(o_i)$ representing the locus of points with common angular characteristics in a starting direction. Any point in the plane can only belong to a polar region, which determines its angular situation with respect to the rest of generator objects in the scene. More specifically, if point p lies in the polar region of object o_i , $p \in \mathcal{P}_E(o_i)$, we know that o_i is the first object found after performing an angular scanning from the horizontal line crossing p in counterclockwise direction. The polar diagram can be computed efficiently using the Divide and Conquer or the Incremental methods, both working in $\Theta(n \log n)$. The strength of using this tessellation as preprocessing is avoiding any angular sweep by locating a point into a polar region in logarithmic time [4].

A KDS is a structure that maintains a certain attribute of a set of continuously moving objects. It consists of two parts: a combinatorial description of the attribute and a set of certificates with the property that as long as the outcomes of the certificates do not change, the attribute does not change. It is assumed that each object follows a known trajectory so that one can compute the failure time of each certificate. Whenever a certificate fails – we call this an event – the KDS must be updated. The KDS remains valid

⋆ This author's work was in part supported by a grant from IPM (N. CS2386-2-01).

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 283–290, 2008.

until the next event. See the excellent survey by Guibas [3] for more background on KDSs and their analysis.

In this paper we use the notion of kinetic data structure to model the dynamic case of polar diagram, i.e we maintain the polar diagram of a set of continuously moving objects in the scene. We Show that our proposed structure meets the main criteria of a good KDS.

The rest of this paper is organized as follows: In section 2.1 we define our kinetic configuration for Polar Diagram, and in section 2.1 we see what happens when the objects move in the plane. In section 2.2 we extend our model for circular objects.

2 Kinetic Polar Diagram

In this section we present a model for kinetic behavior of polar diagram for different situations. Given a set of points moving continuously, we are interested in knowing at all times the polar diagram of the scene.

2.1 Kinetic Configuration

Proof Scheme. For simplicity of disscussions, we assume that our objects are points in 2D. In Section 2.2 we will show that our model is also valid for circular objects.

We claim that if we have the sorted list of objects according to their y-coordinates, and the for each object, its *pivot*, the second object that lies on the polar edge passing the object, we will have a unique polar diagram.

Suppose there are n points in the scene. For our proof scheme, we maintain two kinds of information about the scene: we maintain the vertically sorted list of sites, and for each site its current pivot. As we will show shortly, these data is sufficient for the uniqueness of our polar data, i.e. only if one of these conditions change, the polar structure of the scene will change.

So we will have two kinds of certificates: $n-1$ certificates will indicate the sorted list of sites. For instance, if the sorted list of sites is $s_{i_0}, s_{i_1}, \ldots, s_{i_{n-1}}$, we need the certificates $s_{i_0} < s_{i_1}, s_{i_1} < s_{i_2}, \ldots, s_{i_{n-2}} < s_{i_{n-1}}$.

For stating the pivot of each object, we need n more certificates, each indicating a site and its pivot in polar diagram. In total, our proof scheme consists of $2n-1$ certificates.

Events and Event Handling. Once we have a proof system, we can animate it over time as follows. As stated before, each condition in the proof is called a certificate. A certificate fails if the corresponding function flips its sign. It is also called an event happens if a certificate fails. All the events are placed in a priority queue, sorted by the time they occur. When an event happens, we examine the proof and update it. An event may or may not change the structure. Those events that cause a change to the structure are called *exterior events* and those not *interior events*. When the motion of an object changes, we need to reevaluate the failure time of the certificates that involve that object (this is also called *rescheduling*.).

As there are two kinds of certificates in our proof scheme, it is obvious that there must be two kinds of event:

- **pivot event**, when three objects, which one of them is pivot of another one, become collinear.
- **horizontal event**, when two objects have a same y-coordinate (have a same horizontal level)

In the former case, we must update the certificates relating to sorted sequence of two neighbor points, which is at most three certificates (two, if one of the points is a boundary point, i.e. top most or button most points). In the latter case, one certificate becomes invalid and another certificate (indicating the new pivot of the site) is needed. As we will show, other certificates will remain still.

Lemma 1. *When an event is raised, the objects above the object(s) which raised the event do not change their polar structures.*

Proof: From the incremental method used for the construction of the polar digram of a set of points [4] we know that there is no need to know about the state of objects below a site to determine its pivot object. We can also say that an angular sweep that starts from the horizontal direction would never intersect any objects below this initial horizontal line (by definition, the top most site has no pivot). □

Pivot event

First, we consider the simplest case, i.e. when the lowest object is moving. Figures 1 and 2 show these cases, where s_2 is moving. In Figure 1, s_0 is the pivot of s_2. While s_2 is moving left, the line segment s_0s_2 is coinside with the site s_1 (note that there may be other sites between s_0 and s_2, but we are only interested in s_1). At the moment that three sites s_0, s_1, and s_2 become collinear, the s_1 will occlude s_0 from s_2 and it no longer can be its pivot. From now on, s_1 becomes the new pivot of s_2. Similarly, in Figure 2, s_1 is the pivot of moving site s_2. When three sites s_0, s_1, and s_2 become collinear (again, there may be other sites between each pair of these sites, but we are not interested in them), s_2 needs to change its pivot which becomes s_2.

As we assumed that no other object other than s_2 is moving, form lemma 1 we know that there will be no change in other objects, so at this event, only one certificate becomes invalid and it must be replaced by another certificate indicating the new pivot of the moving object. It is clear that upon occurring this event, the processing of the event and changing of proof scheme can be done in $O(1)$ and $O(\log n)$, respectively (we need to find the corresponding certificate in the certificates list).

Now we see what happens to the second lowest site (see Figures 3 and 4, where s_2 is moving right). In Figure 3, s_1 is the pivot of s_2, and also the pivot of the lower site s_3. While moving, there will be a time that s_2 occlude the lower site s_3 from its pivot. In Figure 3 it is when the sites s_1, s_2 and s_3 become collinear. At this time, although there

Fig. 1. A pivot event. As s_2 moves left, s_0, s_1 and s_2 become collinear

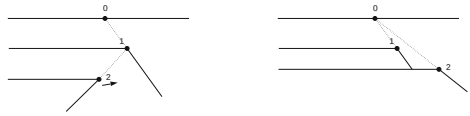


Fig. 2. A pivot event. As s_2 moves right, s_0, s_1 and s_2 become collinear.

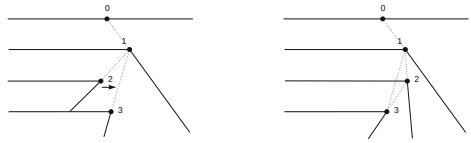


Fig. 3. While moving, s_2 can change the pivot of each of its below sites by occluding their initial pivots

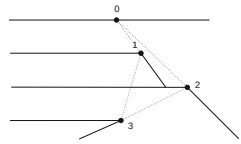


Fig. 4. For each moving site, there is one pivot event when its own pivot will change

is no change in polar structure of moving site s_2, there is a change in the lower site s_3, and we must update the proof scheme accordingly. If s_2 continues its motion, there will be a pivot event (see Figure 4) that its polar structure is changing.

Lemma 2. *The changes in the structure of a site caused by moving an above object, would not cause any other changes in other sites.*

Proof: The Structure of each site is determined by the first site that encountered by an angular sweep. As we assumed that no other objects is moved, this encountered site would not change. □

From above discussions, we can deduce that if a site is moving in the scene and there are k other sites below it, there can be up to k pivot events changing the structure of below sites, and one pivot event changing its own structure. Each of these events can be processed in $O(1)$ time and the change in proof scheme can be done in $O(\log n)$.

Horizontal event

In these events, one of the situations of Figures 5 and 6 will happen. As we can see, only one of the sites will change its pivot (set it to the third object). This change of configuration is equal to changing three or four certificates in proof scheme: one for a change in one of the site's pivot, and three or two for change in vertical order of sites.

Now we show that no more changes is needed. Assume that in a small interval before and after the horizontal event, no other pivot events would occur. From lemma 1 we know that there would be no change in the above objects. What about the below sites? We can see that for a change in the pivot of a site, there must be an occlusion between

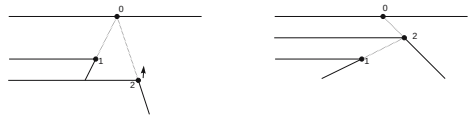


Fig. 5. When two sites s_1 and s_2 lay on a same horizontal level, a horizontal event is occurred and the polar structure will change

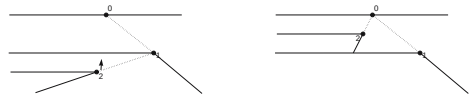


Fig. 6. In a horizontal event, only one of the sites will change its pivot

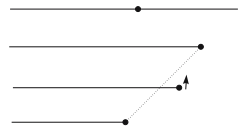


Fig. 7. Only upon occurring a pivot event the structure of other sites will change

the sites and its previous pivot, and it means that three sites must lay on a same line, i.e. we need a pivot event (see Figure 7).

Theorem 1. *Each of the events in kinetic polar diagram of a set of points takes* $O(\log n)$ *time to process and causes has* $O(1)$ *changes in proof scheme.*

Proof: For horizontal events, we need to update at most three certificates, we just need to find these certificates in the proof scheme and replace them with the new ones, which takes $O(\log n)$ time. We also need to update one pivot certificate with the same cost. The same thing is holds for pivot events, which we need to find and update $O(1)$ pivot certificates. □

Theorem 2. *The initial event list can be built in* $O(n \log n)$ *time, using a suitable event queue.*

Proof: As there are $O(n)$ certificates in our proof scheme, and for each moving object. we can find the first certificate that it will violates by a simple $O(\log n)$ search, the proof is straightforward. □

2.2 Circular Objects

For circular objects, we use a similar approach to that of previous section about the line segments. For our proof scheme, we maintain a sorted list of all $2n$ North and South poles. It can be done by $2n - 1$ certificates. Also, for each oblique polar edge, we add a certificate, denoting its main object and its pivot. As there may be up to $3(n + 1) - 6$ such edges [4], we may have up to $3n - 3$ such certificates. Like the point objects case,

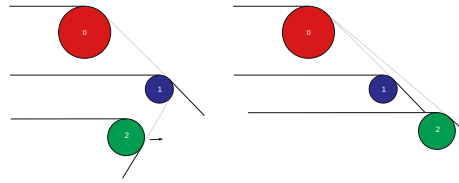


Fig. 8. As three objects s_0, s_1 and s_2 form a tri-tangent, a pivot event will occur

we have two kinds of events upon moving of objects: horizontal events and pivot events. As we will see, while handling these events, there might be one other type of change in polar structure which we are not interested in, i.e. as we used a lazy structure for our proof scheme, we do not consider this type of change. This is when a polar edge is occluded by another object in its way.

Pivot event
These events are essentially the same as those for point objects. As we can see in Figure 8, when three objects become tri-tangent, there is a potential pivot event: when one of them is pivot of another one, we have a pivot event. In these events, the object that has its pivot in trio will change its pivot and we need to replace the corresponding certificate in proof scheme with a another one.

Horizontal event
As there are $2n$ poles for n circular objects, the processing of horizontal events are a little different from those of point objects. Figures 9 and 10 shows the cases where two different pole types lay on a same horizontal level. As we can see, in the case of Figure 9, a new polar edge from a South pole appears, and in case of Figure 10, a previous present polar becomes occluded. As we said before, we take non of these changes in polar structure in our proof scheme, and we only need to update certificates corresponding to the vertical order of poles.

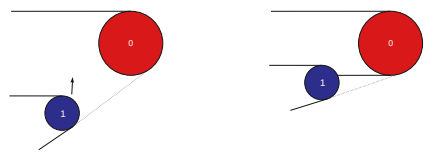


Fig. 9. A horizontal event. A polar edge from a South pole will appear.

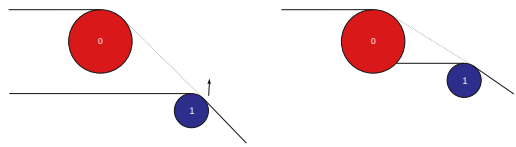


Fig. 10. A horizontal event. A polar edge will be occluded.

Another type of horizontal event occurs when two pole of the same kind (North or South) lay on a horizontal line (Figures 11). Apart from appearing or occluding of polar edges, there might be another change in polar structure. In these cases, an oblique edge can appear (Figure 11) or disappear. So we need to add or remove the corresponding certificates indicating the oblique polar edge.

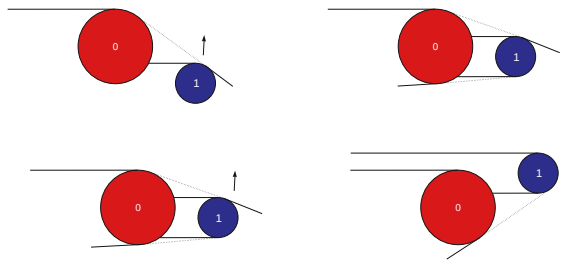


Fig. 11. Horizontal events. (a) A polar edge from a South pole and an oblique edge will appear. (b) A polar edge will no be occluded anymore.

From above discussions we can deduce the following proposition.

Proposition 1. *Each of the events in kinetic polar diagram of a set of circles takes* $O(\log n)$ *time to process and it has* $O(1)$ *changes in proof scheme.*

2.3 KDS Evaluation

Evaulation of a good kDS depends on some properties [2]. Here we consider thease properties in our kinetic model.

Compactness. The size of the proof. The structure clearly takes linear space. As we stated in Section 2.1, for a set of n point objects, the proof scheme consists of $n-1$ certificates for sorted vertical order of objects and n certificates for maintaining the pivots of each object, so in total, our proof scheme have $2n-1$ certificates.

Responsiveness. The time to process an event. $O(\log n)$ for processing an event as there are $O(1)$ certificates need to reschedule. Each reschedule takes $O(\log n)$ time.

Locality. The number of certificates that a single object involves in. Each object is involved in at most three certificates.

Efficiency. The number of events processed. All the events are exterior – the ordering changes once a horizontal event happens, or the pivot of an object changes once a pivot event happens. The number of events is bounded by $O(n^2)$ as any two points can exchange their ordering only constant number of times for constant degree algebraic motions, and any point is a potential candidate for being the pivot of another point.

3 Conclusion and Future Work

In this paper we studied the concept of Polar Diagram, which is a new locus approach for problems processing angles, and KDS, which is a structure that maintains a certain attribute of a set of continuously moving objects among moving objects. We used KDS to model the behavior of a Polar Diagram when our scene is dynamic, i.e. we maintain the polar diagram of a set of continuously moving objects. We showed that our proposed structure meets the main criteria of a good KDS.

Following our defined model for kinetic polar diagram, we can use it in direct applications of polar diagram to maintain the computed attributes. For example, we can use kinetic polar diagram for maintaining the convex hull of a set of moving objects with a very low cost.

References

1. Basch, J., Guibas, L.J., Hershberger, J.: Data structures for mobile data. In: Proc. 8th ACM-SIAM Sympos. Discrete Algorithms, pp. 747–756 (1997)
2. Basch, J.: Kinetic data structures. PhD thesis, Stanford University, Palo Alto, California (1999)
3. Guibas, L.: Kinetic data structure: a state of art report. In: Proc. 3rd Workshop Algorithmic Found Robot., pp. 191–209 (1998)
4. Grima, C.I., Márquez, A., Ortega, L.: A new 2D tessellation for angle problems: The polar diagram. In: Computational Geometry, Theory and Application (2006)
5. Grima, C.I., Márquez, A., Ortega, L.: A locus approach to angle problems in computational geometry. In: Proc. of 14th European Workshop in Computational Geometry, Barcelona (1998)

A Graph Transformation-Based Approach to Formal Modeling and Verification of Workflows

Vahid Rafe and Adel T. Rahmani

Department of Computer Engineering
Iran University of Science and Technology
Tehran, Iran
{rafe,rahmani}@iust.ac.ir

Abstract. This paper proposes a formal approach to modeling and verification of workflows using graph transformation systems. To model workflows, we use UML 2.0 activity diagrams. As this kind of diagram does not possess a precise formal semantics, therefore we propose a formal semantics for dynamic behavior of activity diagrams using graph transformation systems and then we verify them by model checking. To verify workflows, we use our previous approach to model checking of graph transformation systems – using Bogor model checker to verify graph transformation systems-.

Keywords: Workflow Modeling, Graph Transformation, Verification, Model Checking.

1 Introduction

Most of the real-world tasks can be described by or are part of processes, i.e., "series of actions or operations conducing to an end" [1]. These processes are usually complex. Consider a large enterprise organized in several departments, each of them having different responsibilities. A typical task may involve some or all of those departments. Some departments may depend on the work of others; some may work in parallel to increase processing speed and so forth. Business processes of this kind are often referred to as workflows [2].

These workflows play an important role to the success of enterprises: An inefficient or faulty workflow will result in low productivity and may even be lead to serious problems in the processing of tasks. Additionally, as workflows involve large and different organizations, they can not easily be tested or verified on the organizations themselves. It is therefore desirable to develop a model of a workflow, which can then be verified and tested. Beside the complexity described above, as different people with different skills are involved in workflows, this will make the modeling of workflows a difficult task. Hence using of a visual modeling language to overcome these problems is a natural choice.

Since the past decade, Unified Modeling Language (UML) has been a standard modeling language to express models in a software development process and most of people in software engineering community are familiar with that. One big advantage is that UML diagrams are designed to be easily understandable. But the major drawback

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 291–298, 2008.

of UML and similar modeling languages is that they only define syntax for modeling without a precise formal semantics. Formal methods are crucial in automated software engineering. But the problem of formal methods is that they are difficult to be understood by designers because there is a complex mathematics behind them. Therefore; our proposal tends to use UML 2.0 activity diagrams [3] as a syntax notation, while using graph transformation systems [4] as a formal semantics background.

The proposal presented in this paper addresses an approach to modeling workflows using graph transformation systems and then verifying them. At first, we will present a formal semantics for activity diagrams using graph transformation systems. The main motivation for choosing graph transformation among different formalisms is that most of diagrams in software models (e.g. activity diagrams) are graphs. Based on this formalism, designers can model workflows by graph transformation (but syntactically similar to activity diagrams). We use activity diagrams syntax notation to ease understanding the formal -graph transformation based- diagrams. After modeling workflows as a graph transformation system, we use our previous approach to verify graph transformation systems [5]. We verify designed workflows automatically by model checking. We check different properties like reachability, deadlock freeness, safety, and other interesting properties on these obtained formal models.

The paper is organized as follows. Section 2 surveys the related works. Section 3 describes our approach to define a formal semantics for workflow modeling. Section 4 shows our approach to verification of modeled workflows and section 5 concludes the paper.

2 Related Work

There is much research done about formal modeling and verification of workflows using different formal languages. In [6], J. H. Hausmann defines a concept named Dynamic Meta Modeling (DMM) using graph transformation. He extends the traditional graph rules by defining a new concept named rule invocation. In DMM there are two kinds of rules: big-step and small-step rules. Big-step rules act as traditional rules but small-step rules should be invoked by big-step rules. Using these kinds of rules, modeling of complex systems is easier. Hausmann then defines semantics for activity diagrams using concept of DMM. Soltenborn [3] uses DMM and defines semantics for activity diagrams for modeling and verification of workflows. For verification, he uses GROOVE [7], but as GROOVE does not support attributed type graphs, therefore, he changes the rules to be verifiable by GROOVE. He checks deadlock freeness and action reachability properties on the modeled workflows. But against our approach he can not check a specified action. His approach can only show is there any unreachable action in the model or not. Another problem is that his approach can not model events. Also the extension defined by Hausmann can not be modeled directly in existing graph transformation tools.

R. Eshuis [8] uses UML 1.5 activity diagrams to model workflows. Based on Clocked Transition Systems (CTS), he defines a formal semantics for activity diagrams (a statechart-like semantics). He considers control and data flow for modeling and verification of workflows. He defines a property called strong fairness to verify functional requirements of the model. This approach uses NuSMV [9] model checker

to check strong fairness property stated in LTL expression. In contrary to our method, this approach can support dataflow, but it defines a concept named advanced activity diagram that makes it difficult to be used by designers. Furthermore, our approach can check more properties rather than the strong fairness.

3 Modeling Workflows

Our approach focuses mainly on control flow perspective; therefore, for modeling workflows we consider these parts of activity diagrams: *Init* node, *Final* node, *Action* node, *Fork* node, *Join* node, *Merge* node, *Decision* node and *AcceptEvent* node (to support event modeling in workflows). Beside these nodes we use some constraints on the models. These constraints are as following:

1- Each activity diagram must have exactly one *Init* node and one *Final* node.
2- *Init* node has not any incoming edge and *Final* node has not any outgoing edge.
3- Each *Fork* and *Decision* node should have exactly two outgoing edges. Note that it is possible to have these nodes with more outgoing edges by cascading them. Therefore, it has not any restriction on our models.
4- Each *Action*, *Merge*, *Init* and *Join* nodes should have exactly one outgoing edge.
5- The source and target node of each edge should be identical. (There must not be any self-edge in the graphs.)
6- Each *Final*, *Action*, *Fork*, *Decision* nodes should have only one incoming edge.
7- Each *Join* and *Merge* nodes should have exactly two incoming edges. Note that it is possible to have these nodes with more incoming edges by cascading them.
8- Each *Action* node can have some outgoing edges to some different *Accept Event* node and each *Accept Event* node should have exactly one outgoing edge to an *Action* node (this kind of edges is different with other edges).

We have proposed these constraints to have models with precise syntax and it is possible to draw many UML 2.0 activity diagrams by these constructs[1].

The class diagram shown in fig 1 represents a portion of UML 2.0 activity diagrams' metamodel [10]. This metamodel can be easily considered as a type attributed graph. Since UML 2.0 specification, stipulates that activities "use a Petri-like semantics" [3], we will use token-flow semantics in our graphs. Therefore; to show tokens we add an attribute to each node named "token" of type boolean. Fig 2 shows the proposed typed graph for activity diagrams.

Fig 2 shows the designed type graph based on mentioned constraints. This type graph and other parts of proposed graph transformation system are designed in AGG toolset[2]. AGG [11] automatically checks that each host graph (activity diagram and

[1] We do not consider labels or guards on the edges because it has not any effect on our approach to workflow verification.
[2] http://tfs.cs.tu-berlin.de/agg/

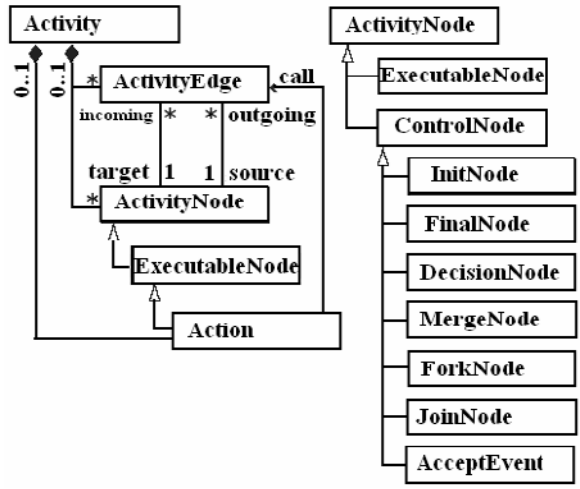


Fig. 1. A small portion of UML 2.0 metamodel [10]

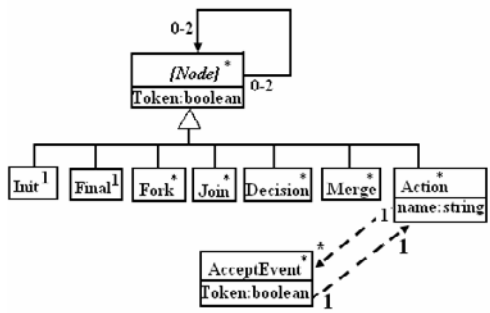


Fig. 2. Proposed type graph of UML 2.0 activity diagram

rules) should be consistent with its type graph and other constraints. Therefore, when we model a workflow in AGG, we are sure it syntactically is consistent with type graph and other constraints.

The proposed type graph consists of one abstract type (i.e. *Node)*, the star (*) sign on the top right corner shows the multiplicity of these nodes in the models. Other nodes (except *AcceptEvent*) have inherited it. It means all nodes have the *Token* field. As it is shown, there are two different kinds of edges, only *Action* nodes can use dashed edges with *AcceptEvent* nodes and vice versa. The multiplicities of edges show the minimum and maximum edges that each node can have as incoming or outgoing edges. This type graph does not satisfy all the above constraints. Therefore, we need some more constraints beside this type graph. For example, based on this type graph, it is possible to have host graphs (activity diagrams) with some *Action* nodes that have not any outgoing edge or incoming edge. In AGG, using *atomic graph constraint* and *formula constraint*, we can define these additional constraints on the model.

After adding other constraints to the graph transformation system, the static part (metamodel) of the proposed formal semantics has been completed. Now we can model workflows directly as host graph in graph transformation system. Fig 3 shows a sample workflow modeled by activity diagram [3]. Dashed region shows the area that the event *"Cancel Order Request"* can be activated. All *Action* nodes in this area can activate this event. This activity diagram can be modeled as the host graph in fig 4. As it is shown, this host graph is consistent with the type graph and constraints. Note that we model workflows directly as a host graph instead of transforming UML activity diagrams to host graphs.

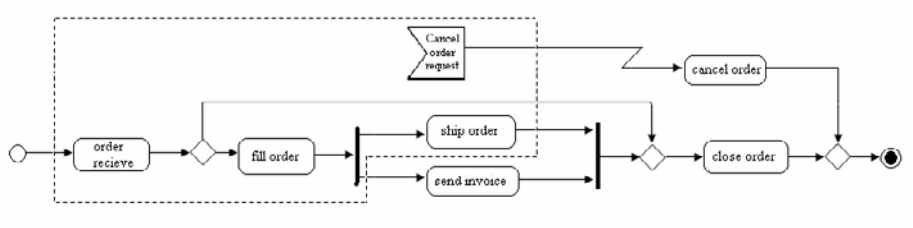


Fig. 3. A sample activity diagram

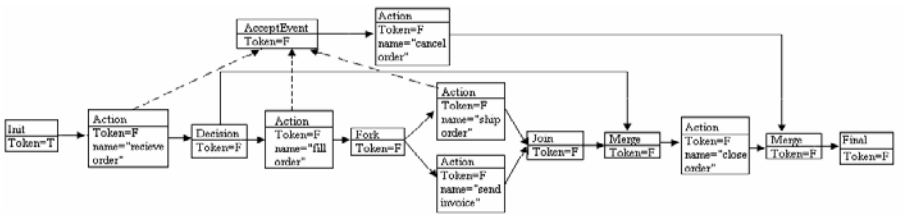


Fig. 4. The sample activity diagram in fig 3 as host graph

Proposed rules show the token flow in the host graph. To define these rules we consider these definitions for token flow:

- In each workflow, at first, only the *Init* node has the token, i.e., the *Token* attribute of *Init* node is true and this attribute is false for other nodes.
- Tokens can not stick on nodes, it means as soon as there is a suitable way, the token should be propagated.
- The flow of tokens will be finished when *Token* attribute of all nodes are true, or when there is not any way for tokens to be propagated (i.e. there is not any rule to be applicable on the model). It means it is possible that token reaches the *Final* node, before reaching some other nodes. Therefore; our rules should be designed in a way that considers this fact.
- If token reaches to a node for twice, we ignore this token (except for Join nodes). Since our verification approach checks that all actions should be reachable and the workflow should be deadlock free, we can ignore recurring tokens. In Join nodes, if exactly two tokens arrive, they can fire a token to their following node.

Based on these definitions and the desired behavior of activity diagrams, we have proposed 24 graph transformation rules as dynamic semantics for the workflow models. Due to the lack of space, we can not explain all them here, but we briefly describe some of them. Fig 5 shows two proposed rules. Rule 1(a) shows the token flow from *Init* node to its following node. The NAC shows that the following node can not be a *Join* node, because the semantics of *Join* nodes are different. The NAC simply depicts that the following node of *Init* should not have any previous node. It means the following node can not be a *Join* or *Merge* node (because there are two kinds of nodes in our type graph that can have two incoming edges). But we only want to prevent of application this rule for *Join* nodes. Therefore; we need another rule for the case that the following node of *Init* is a *Merge* node. Rule 1(b) shows this case.

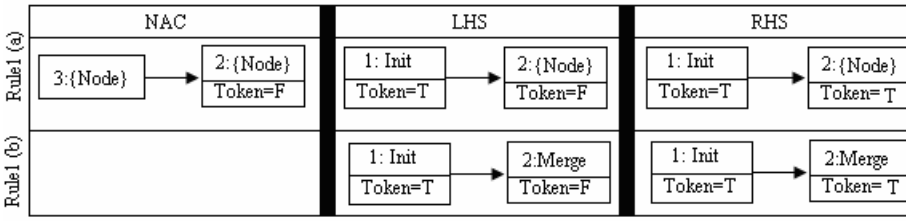


Fig. 5. Two proposed rules to show token flow in workflows

4 Verification of Workflows

To verify designed workflows we use our previous approach to verify graph transformation systems [5]. To verify workflows, we have designed some fixed properties, it means designers can only model the workflow, and then automatically check the properties. In addition, it is possible to define new properties by designers.

One property which should hold for a sound workflow is that it should be deadlock free. For this purpose we should check that for all executions of the workflow, *Final* node is reachable. We can show this property by a rule. Fig 6(a) shows this rule (both LHS and RHS are identical and this rule has not any NAC). The rule only states that the value of token attributes for all *Final* nodes in the model should be false (as we have only one *Final* node in workflow models, the property will be translated only for this node). Suppose the name of this rule is property1. The LTL expression □

(property1 → ◊(¬property1)) shows this property. It is true if in every possible execution there is a state in a path in which *property1* is satisfied (the token attribute of *Final* node is false)␣and then eventually there is a state in the postfix of that path, in which the token attribute of the *Final* node is true (i.e. the *Final* node is reachable). As an example, fig 6(b) shows a workflow which contains a deadlock. We have used UML 2.0 activity diagram notation to show this diagram to be more understandable. In this diagram, there is a *Join* node immediately after *Init* node and it prevents token to be propagated from *Init* node. Therefore; token never reaches to *Final* node and it causes a deadlock.

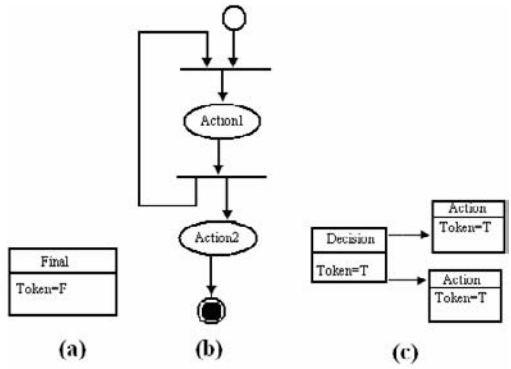


Fig. 6. (a) A rule to show property1; (b) A faulty workflow which contains a deadlock; (c) A rule to show property2

If we want to validate our approach by some benchmarks, we can use model checking for this purpose. For example we can check is there any execution for a workflow which both *Action* nodes after a *Decision* node have the token. Fig 6(c) shows a rule for this purpose. The LTL expression ¬□(¬property2) states this property.

5 Conclusion and Future Work

This paper presents an approach to model and verification of workflows. We use UML 2.0 activity diagrams as syntax notation for modeling, but to have an automated approach to modeling and verification, we need a formal language; while UML lacks the formal semantics. Hence we use graph transformation systems as a formal background. Using token flow semantics, we define a number of rules and a type graph and each workflow is modeled by a host graph. Then we verify the designed workflow by model checking. To model checking of graph transformation systems we use our previous approach (i.e. translation graphs and properties to BIR and then model checking by Bogor[3] [12]). Using this solution to verify graph transformation systems has a good flexibility to checking different properties on the modeled workflows and this point highlighted our approach among other existing practices.

However, further research is required to model other required elements (e.g. Exceptions and Action calls). We have a plan to model these elements and define semantics for them as some graph transformation rules.

Acknowledgments

We would like to thank professor Luciano Baresi (Politecnico di Milano) for his valuable discussions and ideas. We are also grateful to Dr. Paola Spoletini (Politecnico di Milano) for her ideas about model checking and verification phase.

[3] http://bogor.projects.cis.ksu.edu/

References

1. Encyclopedia Britannica. Dicitionary and Thesaurus, http://www.britannica.com/
2. Soltenborn, C.: Analysis of UML Workflow diagrams with dynamic Meta Modeling Techniques, Master's Thesis, University of Paderborn, Germany (2006)
3. OMG Unified Modeling Language: Superstructure (final adopted spec, version 2.0), Technical report, Object Management Group (2003).
4. Baresi, L., Heckel, R.: Tutorial Introduction to Graph Transformation: A Software Engineering Perspective. In: Corradini, A., Ehrig, H., Kreowski, H.-J., Rozenberg, G. (eds.) ICGT 2002. LNCS, vol. 2505, pp. 402–429. Springer, Heidelberg (2002)
5. Baresi, L., Rafe, V., Spoletini, P., Rahmani, A.T.: An Efficient Model Checking Approach for Graph Transformation Systems. In: Proceedings of 3th Int. Workshop on Graph Transformation for Verification and Concurrency (GT-VC 2007) (2007)
6. Hausmann, J.H.: Dynamic Meta Modeling: A Semantics Description Technique for Visual Modeling Languages, Ph.D. Thesis, University of Paderborn, Germany (2005)
7. Rensink, A.: The GROOVE Simulator: A Tool for State Space Generation. In: Pfaltz, J.L., Nagl, M., Böhlen, B. (eds.) AGTIVE 2003. LNCS, vol. 3062, pp. 479–485. Springer, Heidelberg (2004)
8. Eshuis, R.: Semantics and Verification of UML Activity Diagrams for Workflow Modelling, Ph.D. Thesis, University of Twente, Netherlands (2005)
9. Cimatti, A., Clarke, E., Giunchiglia, F., Roveri, M.: NuSMV: A new symbolic model checker. International Journal on Software Tools for Technology Transfer 2(4), 410–425 (2000)
10. Störrle, H.: Semantics of Control-Flow in UML 2.0 Activities. In: Proc. of IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC (2004)
11. Beyer, M.: AGG1.0 – Tutorial. Technical University of Berlin, Department of Computer Science (1992)
12. Robby, Dwyer, M., Hatcliff, J.: Bogor: An Extensible and Highly-Modular Software Model Checking Framework. In: Proc. of the 9th European software engineering Conffference, pp. 267–276 (2003)

Efficient Parallel Routing Algorithms for Cartesian and Composition Networks

Marzieh Bakhshi[1], Saeedeh Bakhshi[1,2], and Hamid Sarbazi-Azad[1,2]

[1] Sharif University of Technology, Tehran, Iran
[2] IPM School of Computer Science, Tehran, Iran
m_bakhshi@ce.sharif.edu, {bakhshi,azad }@ipm.ir

Abstract. The Cartesian and Composition products are two well known graph products, also applied to interconnection networks area. The Cartesian and Composition network products possess great characteristics of fault resilience according to their high connectivity. In this paper, we study the existence and construction of parallel routing paths in these two well-known product networks. To prove the existence of certain number of parallel paths in these product networks, we need to compute their connectivity. By assuming the availability of certain number of faulty nodes, we propose some new shortest-path parallel routing algorithms. These algorithms can be used both in faulty networks and to route different data from different paths to a specified target node. We also study the concept of container length and wide diameter of the two mentioned product networks. Comparison between wide diameters and diameters of the two networks exhibit the efficiency of these product networks in constructing node disjoint paths.

Keywords: Parallel routing, Cartesian product, Composition product, Fault tolerant routing, Connectivity.

1 Introduction

The Cartesian and composition products are well-known operations to create new networks defined on some factor graphs. Several well-known networks, e.g. hypercubes, tori, and meshes, are instances of these networks. Different aspects of product networks have been studied in the literature. The interest in product networks is due to their attractive mathematical structure, as well as their increased networking power and versatility. In this paper, we study the connectivity and node-disjoint parallel path routing algorithms in such networks. The rest of the paper is organized as follows: In section 2, we formally state some definitions, which are used through the paper; Cartesian and Composition networks are introduced in this section. Parallel routing algorithm of Cartesian networks and its wide diameter is discussed in section 3. In section 4, we propose an algorithm for parallel routing in Composition networks and then obtain its wide diameter. Finally, the paper is concluded in section 5.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 299–307, 2008.

2 Definitions and Preliminaries

If G is a connected graph and u and v are two nodes of G, a (u,v)-*container* in G denoted by $C(u,v)$ is a set of node-disjoint paths between u and v [8]. The number of paths in $C(u,v)$ is called the *width* of $C(u,v)$ denoted by $w(C(u,v))$. The length of longest path in $C(u,v)$ is called the length of $C(u,v)$ denoted by $l(C(u,v))$. A $C(u,v)$ is called the *best* if its length is minimum. We use $C_x(u,v)$ to denote a $C(u,v)$ with width x, and $C_x^*(u,v)$ to denote the best $C_x(u,v)$, where $x \geq 1$. The *x-wide distance* between u and v is defined as $l(C_x^*(u,v))$, where $w(C_x^*(u,v)) = x$ and $x \geq 1$ [8]. The *x-wide diameter* of G, denoted by $wd_x(G)$, is defined as the maximum x-wide distance between two arbitrary nodes of G [8].

The Cartesian product graph $G = G_1 \times G_2$ of factor graphs G_1 and G_2 with disjoint point sets V_1 and V_2 and edge sets X_1 and X_2 is the graph with point set $V_1 \times V_2$; $u = (u_1, u_2)$ is adjacent to $v = (v_1, v_2)$ if $u_1 = v_1$ and u_2 is adjacent to v_2, or if $u_2 = v_2$ and u_1 is adjacent to v_1. For two given graphs G and H, the *composition* of these two graphs, denoted by $G[H]$, is the graph with vertex set $V(G) \times V(H)$ where (u_1, v_1) is adjacent to (u_2, v_2) if either u_1 is adjacent to u_2 in G, or $u_1 = u_2$ and v_1 is adjacent to v_2 in H. We define E_K as the number of edges, and V_K as the number of vertices in graph K, and assume $d_K(u) : u \in V_K$ is the degree of vertex u in graph K, and D_K is the diameter of graph K. Note that both G and H are connected graphs.

3 Parallel Routing in Cartesian Network

Lemma1 [2]. $diam(G_1 \times G_2) = diam(G_1) + diam(G_2)$ and $\kappa(G_1 \times G_2) \geq \kappa(G_1) + \kappa(G_2)$. □

Theorem 1. $l\left(C^*_{\kappa(G_1)+\kappa(G_2)}(u,v)\right) \leq l\left(C^*_{\kappa(G_1)}(u_y, v_y)\right) + l\left(C^*_{\kappa(G_2)}(u_x, v_x)\right)$ for every two distinct nodes u and v of $G_1 \times G_2$.

Proof. To prove the theorem, it is enough to find $\kappa(G_1) + \kappa(G_2)$ disjoint paths between u and v whose lengths are not greater than $l\left(C^*_{\kappa(G_1)}(u_y, v_y)\right) + l\left(C^*_{\kappa(G_2)}(u_x, v_x)\right)$. Suppose that $u = (u_x, u_y)$ and $v = (v_x, v_y)$.

Case 1: $u_x = v_x$. In this case, $\kappa(G_1)$ disjoint paths exist in G_1 between u_y and v_y. Also there are at least $\kappa(G_2)$ vertices adjacent to u_y existed in the rest of $V(G_2) - 1$ graph (each isomorphic to G_1). Since there are at least $\kappa(G_2)$, G_1^i-subgraphs connected to $G_1^{u_x}$-subgraph, the containers can be formed as follows: $u = (u_x, u_y) \xrightarrow{\kappa(G_1) \text{ containers in } G_1^x} v = (v_x, v_y)$

$u=(u_x,u_y)\rightarrow(u_{i_k},u_y)\xrightarrow{\kappa(G_1)\text{ containers in }G_1^{i_k}}(u_{i_k},v_y)\rightarrow(v_x,v_y)$ for $(u_x,u_{i_k})\in E(G_2)$ and $1\le k\le\kappa(G_2)$. Hence, in this case, there are $\kappa(G_1)(\kappa(G_2)+1)$ disjoint paths with length of at most $l\left(C^*_{\kappa(G_1)}(u_y,v_y)\right)+2$.

Case 2: $u_x\neq v_x$. In this case, we label the $\kappa(G_1)$ containers in $G_1^{u_x}$ with $C_i(u_y,v_y)$, $1\le i\le\kappa(G_1)$; the vertices that are presented in each of containers are shown as $c_j^i:1\le j\le l(C_i)$. We can construct the paths as follows: From $u=(u_x,u_y)$ to (u_x,v_y), there are $\kappa(G_1)$ disjoint paths. Consider the disjoint paths between u_y and v_y. We can construct $\kappa(G_1)$ disjoint paths by going from $u=(u_x,u_y)$ to $(u_x,c_i^2):1\le i\le\kappa(G_1)$. After reaching to $(v_x,c_i^2):1\le i\le\kappa(G_1)$, we can continue the specified disjoint paths in G_1 until reaching to $v=(v_x,v_y)$. The paths are disjoint and of length at most $l\left(C^*_{\kappa(G_1)}(u_y,v_y)\right)+l\left(C^*_{\kappa(G_2)}(u_x,v_x)\right)$. The other $\kappa(G_2)$ containers can be built by going from $u=(u_x,u_y)$ to the same node in another G_1-subgraph and continuing this process until we reach an arbitrary G_1-subgraph, say $G_1^i:i\neq u,v$. In $G_1^i:i\neq u,v$, we go through one of $\kappa(G_1)-1$ containers (one of them used in the previous disjoint paths) until reaching to node (i_x,v_y). From that node, we can continue the path in G_2 (it was not used yet) to the target node $v=(v_x,v_y)$. We can build $\kappa(G_2)$ such containers in the product graph, as well. It is possible by using a path through one of G_1-subgraphs in the routing so that none of the G_1-subgraphs are repeated more than once. These paths are of length $l\left(C^*_{\kappa(G_1)}(u_y,v_y)\right)+l\left(C^*_{\kappa(G_2)}(u_x,v_x)\right)$. □

Algorithm Cartesian-Parallel-Routing

Input: Parallel routing in G_1: PR_{G_1}, Parallel routing in G_2: PR_{G_2}, Source vertex $s=(s_{G_1},s_{G_2})$, Destination vertex $t=(t_{G_1},t_{G_2})$.

Output: Set C containing all disjoint paths that connect s to t.

Begin

1 Make an empty set *temp*.

2 If $s_{G_1}=t_{G_1}$ then

2.1 Route from $s=(s_{G_1},s_{G_2})$ to $t=(t_{G_1},t_{G_2})$ using PR_{G_1} and add the output to *C*.

2.2 For all $u=(u_{G_1},u_{G_2})$ that $(s,u)\in E_{G_1\times G_2}$ AND $u_{G_1}\neq t_{G_1}$

2.2.1 Route from s to u by traversing an edge.

2.2.2 Route from *u* to $w=(u_{G_1},t_{G_2})$ using PR_{G_1} and add the output to set *temp*.

2.2.3 Route from w to t by traversing an edge.

2.2.4 For every member of *temp*, attach *s* to the beginning and *t* to the end, and add it to *C*.

2.2.5 End for.

2.3 Return

3 If $s_{G_1}\neq t_{G_1}$ and $s_{G_2}=t_{G_2}$ then

3.1 For all $u=(u_{G_1},u_{G_2})$ that $(s,u)\in E_{G_1\times G_2}$ and $u_{G_1}=t_{G_1}$

3.1.1 Route from s to u by traversing an edge.

3.1.2 Route from *u* to $w=(t_{G_1},u_{G_2})$ using PR_{G_2} and add the output to set *temp*.

3.1.3 Rout from w to t by traversing an edge.

3.1.4 For every member of *temp*, attach s to the beginning and t to the end, and add it to C.

3.1.5 End for.

3.2 Route from s to t using PR_{G_2} and add the output to set C.

3.3 Return.

4 If $s_{G_1}\neq t_{G_1}$ and $s_{G_2}\neq t_{G_2}$ then

4.1 Route from *s* to $w=(s_{G_1},t_{G_2}-1)$ using RP_{G1} and add the output to set *temp*.

4.2 Route from w to $x=(t_{G_1},t_{G_2}-1)$ using RP_{G2} and attach the output to the corresponding member of *temp*.

4.3 Route from *x* to *t* by traversing an edge, attach *t* to the end of each member of *temp* and add *temp* to set *C*.

End.

Theorem 2. $wd_{\kappa(G_1)+\kappa(G_2)}(G_1\times G_2)=wd_{\kappa(G_1)}(G_1)+wd_{\kappa(G_2)}(G_2)$.

Proof. Since we used the shortest paths in constructing the containers, and by the use of wide diameters of the factor graphs, the theorem directly follows. □

Considering lemma 1 and theorem 2, we can conclude that Cartesian products are efficient in routing disjoint paths in the presence of faults.

4 Parallel Routing in Composition Network

Lemma 2. If G is complete and H is not, the diameter of *G[H]* would be 2, otherwise the diameter of *G[H]* equals that of G, i.e. D_G.

Proof. First, we label different subgraphs of *G[H]*, G_i $\forall i:1\leq i\leq V_H$, which are isomorphic to G. We construct a path between two arbitrary nodes *u* and *v* in $G[H]$, and then measure the longest path. There are two cases to consider.

Case 1. Both vertices *u* and *v* are in the same G_i subgraph: By the definition, the longest path between two nodes in G_i is equal to D_G. We prove that even if we jump to another subgraph, the length of the longest path can not be less lower than D_G. Consider the first vertex *x* from which you jump out of G_i containing *u* and *v*. From the definition, we conclude that if there is a vertex in G_k $\forall\, k \neq i$, label of which in G is *w*, then there exists a vertex adjacent to *x* in G_i, that the label in G is also *w*. Thus, we can substitute the jump to another subgraph by traveling to vertex *w* in G_i itself, that produces same length or shorter path to *v*; also if we consider the edges produced from the condition "$u_1 = u_2$ and v_1 is adjacent to v_2" in *G[H]*, using these edges we can only reach to another *w*-labeled vertex in some other G_K, which is in the same situation as our previous *w*-labeled vertex was. This means that using these edges may just lengthen the path or produce the same length path, but not a shorter one.

Case 2. Vertex u is in G_i and vertex v is in G_j, where $i \neq j$. Here, there are two other cases:

a) u and v have different labels in G. Find v 's label in *G*, and then find the vertex with the same label in G_i and name it *x*. now consider the shortest path between u and *x*. While traveling through the vertices of this path, there do exist edges between *y* (which is an inner vertex of the path not equal to u or *x*) and vertices in G_j which have the same label as the adjacent vertices of *y* in *G* (according to the definition). Because the shortest path includes *y* as one of its vertices, we can go to the next vertex after *y* in G_j, and then pass through the same-labeled vertices in G_j as the ones in the path in G_i. In this process, we constructed a shortest path, and just the same as what we said for the proof in part 1, this path is the shortest one.

b) u and v have the same labels in *G*: in this case, there are two conceptions:

i. G_i and G_j are two subgraphs which are in the place of two adjacent vertices of *H*. Hence, u and v are adjacent and the route between them consists them.

ii. G_i and G_j are two subgraphs which are in the place of two disjoint vertices of *H*. Hence, u and v are not adjacent, and the length of the shortest path between them is equal or greater than 2. If from *u*, we go to an adjacent vertex named *t*, in another subgrap G_k $\forall\, k \neq i \neq j$, then we can travel from *t* to *v*, because *u* and *v* have the same label in *G* implying that they have exactly the same set of neighbors.

From **a** and **b.i**, we have $D_{G[H]} = D_G$ and from **b.ii**, we have $D_{G[H]} = 1$ or 2. Therefore, it is clear that if $D_G > 1$ then $D_{G[H]} = D_G$, but if $D_H > 1$ and $D_G = 1$, $D_{G[H]} = 1$ or 2, otherwise $D_{G[H]} = 1$. □

Lemma 3. In the composition product graph *G[H]*, the vertices which are to be removed in order to make the graph disconnected, have to be chosen from different *G*-subgraphs of *G[H]*.

Proof. We use contradiction to prove the lemma. Suppose we eliminate no vertex from a subgraph, say G_m. As derived from the definition, there exist edges between vertices in G_m and every other subgraph (vertices of G_m with the same-labeled vertices as their adjacent ones in other subgraphs). Since *G* is connected, G_m is connected too; so by only considering the mentioned edges, the whole graph is connected. Therefore, the assumption is incorrect, implying that we have to choose vertices from all of the *G*-subgraphs. □

Lemma 4. In the composition product graph *G[H]*, in order to make the graph disconnected, it is required to make all the *G*-subgraphs disconnected.

Proof. Again we use contradiction to prove the lemma. Suppose we leave at least one subgraph G_m connected. Because G_m is connected, it follows the fact explained in the proof of lemma 1, even if we remove some vertices from G_m. Hence, we have to make G_m disconnected. □

Now, we obtain the vertex connectivity of the composition graph. The result is stated in the following theorem.

Theorem 3. The vertex connectivity degree of *G[H]* is $\kappa_{G[H]} = \begin{cases} \kappa_G \times V_H & \text{if } (\kappa_G < V_G - 1) \\ \kappa_G \times V_H + \kappa_H & \text{else} \end{cases}$ where κ_G, κ_H are the vertex connectivity of graphs *G* and *H* respectively.

Proof. By lemma 3 and lemma 4, it can be concluded that in order to make *G[H]* disconnected we have to make every subgraph of it disconnected. □

Lemma 5. For every two distinct vertices $u = (g_1, h_1)$ and $v = (g_2, h_2)$ in *G[H]*, we have $l\left(C^*(u,v)\right) \le WD_G$.

Proof. There are 3 cases. In each case, we construct a container for *G[H]*.

Case 1. $g_1 \neq g_2, h_1 = h_2$ meaning that *u* and *v* are in the same *G*-subgraph. Assume their subgraph is G_1. For *G*, we have a container with length WD_G. Consider the disjoint paths in the *G*-container. Suppose the sequence of vertices in one of them is as $u, (g_1, b_1), ..., (g_1, b_k), v$. In order to make more disjoint paths, from *u* we can go to (g_2, b_1) (apparently from the composition product definition, they are adjacent), and then travel through the same path but in subgraph G_2. We can use this method using all the subgraphs and make a container for *G[H]* with length $\kappa_G \times V_H$ (we have V_H G-subgraphs and in each of them κ_G disjoint paths). Also if *G* is a complete graph, this is also true. Because there do exist paths with length of more than 1 and for simulating those with length 1 in other containers, we can go from u to the same labeled

one as v in that subgraph and then go to v, which makes a path of length 2, that is equal or less than the length of the container for *G*. Hence, we can say that the length of the container for *G[H]* is less than or equal to WD_G.

Case 2. $g_1 = g_2, h_1 \neq h_2$, meaning that *u* and *v* are identical in the original *G* graph, but are in different *G*-subgraphs. Assume u 's subgraph as G_1 and v 's subgraph as G_2. Each of u and v has at least δ_G neighbors in one *G*-subgraph. So, in each subgraph, we can go from u to at least δ_G vertices and then go to v. By this method, we can have more than (or equal to) $\delta_G \times V_H$ disjoint paths of length 2, which is less than or equal to WD_G.

Case 3. $g_1 \neq g_2, h_1 \neq h_2$, meaning that *u* and *v* are different in the original *G* graph and are also in different subgraphs. Assume u 's subgraph as G_1 and v 's subgraph as G_2. The way of constructing the container is like that of in case 1. That is we can travel through the same path in every subgraph and when reached to (g_i, b_k) in subgraph G_i, we go to v. □

Theorem 4. $WD_{G[H]} = WD_G$.

Proof. as a consequence of lemma 5, we have $WD_{G[H]} \leq WD_G$, and since there are vertices, for which WD_G stands, $WD_{G[H]} = WD_G$. □

The pseudo code of the parallel algorithm for the composition product network is as follows.

Algorithm Composition-Parallel-Routing

Input: Parallel routing algorithm in G : PR_G, Source vertex $s = (s_G, s_H)$, Destination vertex $t = (t_G, t_H)$.

Output: Set C containing all disjoint paths that connect s to t.

Begin

1 Make an empty set $temp$.

2 If $s_H = t_H$ and $s_G \neq t_G$ then

2.1 Route from $s = (s_G, s_H)$ to $t = (t_G, t_H)$ using PR_G and add the output to C.

2.2 For $i = 1$ to $i = V_H$; $i \neq s_H$ AND $i \neq t_H$

2.2.1 Route from $s = (s_G, s_H)$ to $(s_G + 1, i)$ by traversing an edge.

2.2.2 Route from $(s_G + 1, i)$ to $(t_G - 1, i)$ using PR_G and add the output to set $temp$.

2.2.3 Route from $(t_G - 1, i)$ to $t = (t_G, t_H)$ by traversing an edge.

2.2.4 For every member of *temp*, attach s to the beginning and t to the end, and add it to C.
2.2.5 End for.
2.3 Return
3 If $s_H \neq t_H$ and $s_G = t_G$ then
3.1 For all u that $(s,u) \in E_{G[H]}$ AND $u \neq t$
3.1.1 Route from s to u by traversing an edge.
3.1.2 Route from u to t by traversing an edge.
3.1.3 Add sut sequence to C.
3.1.4 End for.
3.2 Return
4 If $s_H \neq t_H$ and $s_G \neq t_G$ then
4.1 For $i = 1$ to $i = V_H$;
4.1.1 Route from $s = (s_G, s_H)$ to $(s_G + 1, i)$ by traversing an edge.
4.1.2 Route from $(s_G + 1, i)$ to $(t_G - 1, i)$ using PR_G and add the output to set *temp*.
4.1.3 Route from $(t_G - 1, i)$ to $t = (t_G, t_H)$ by traversing an edge.
4.1.4 For every member of *temp*, attach s to the beginning and t to the end, and add it to *C*.
4.1.5 End for.
4.2 Return
End

From lemma 2 and theorem 4, it can be seen that the wide diameter of composition network follows the same rule as its diameter.

5 Conclusion

In this paper, we studied the container length, wide diameter, and parallel routing algorithm for two famous product networks (Cartesian and composition products). In both cases, the wide diameters follow the same rules as their diameters of factor graphs. These results showed that both networks can well tolerate faults. It is because in the Cartesian product the wide diameter is the sum of its factor graphs wide diameters and in the composition product network the wide diameter of the product network equals that of the first factor network.

References

1. Efe, K., Fernandez, A.: Products of networks with logarithmic diameter and fixed degree. IEEE Trans. on Parallel and Distribute Systems 6, 963–975 (1995)
2. Fernandez, A., Leighton, T., Lopez-Presa, J.L.: Containment properties of product and power graphs. Electronic Notes in Discrete Mathematics 7, 1–4 (2001)
3. Leighton, T.: Introduction to parallel algorithms and architectures: Arrays, Tress and Hypercubes. Morgan Kaufmann, San Mateo (1992)

4. Bondy, J.A., Murty, U.S.R.: Graph Theory with Applications. North-Holland, NY (1976)
5. Dietzfelbinger, M., Madhavapeddy, C., Sudborough, I.H.: Three disjoint path paradigms in star networks. In: Proceedings of the IEEE Symposium on Parallel and Distributed Processing, pp. 4000–4066 (1991)
6. Liaw, S.C., Chang, G.J.: Generalized diameters and Rabin numbers of networks. Journal of Combinatorial Optimization 2(4), 371–384 (1999)
7. Ishigami, Y.: The wide-diameter of the n-dimensionml toroidal mesh. Networks 27, 257–266 (1996)
8. Hsu, D.F.: On container width and length in graphs, groups, and networks. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Science E77-A(4), 668–680 (1994)

A Naïve Bayes Classifier with Distance Weighting for Hand-Gesture Recognition

Pujan Ziaie, Thomas Müller, Mary Ellen Foster, and Alois Knoll

Technical University of Munich,
Dept. of Informatics VI, Robotics and Embedded Systems,
Boltzmannstr. 3, DE-85748 Garching, Germany
{ziaie,muelleth,foster,knoll}@cs.tum.edu

Abstract. We present an effective and fast method for static hand gesture recognition. This method is based on classifying the different gestures according to geometric-based invariants which are obtained from image data after segmentation; thus, unlike many other recognition methods, this method is not dependent on skin color. Gestures are extracted from each frame of the video, with a static background. The segmentation is done by dynamic extraction of background pixels according to the histogram of each image. Gestures are classified using a weighted K-Nearest Neighbors Algorithm which is combined with a naive Bayes approach to estimate the probability of each gesture type.

Keywords: Image Processing, Gesture Recognition, K-Nearest Neighbors, Nave Bayes, Classification, Human-robot interaction.

1 Introduction

When humans interact with one another – and with artificial agents – they make extensive use of a range of non-verbal behavior in addition to communicating via speech. Processing and understanding the non-verbal parts of human communication are crucial to supporting smooth interaction between a human and a robot.

We concentrate on the task of *hand-gesture recognition*: recognizing and classifying the hand shapes and motions of a human user in the context of a cooperative human-robot assembly task. Hand gestures play an important role in this type of interaction, both as an accompaniment to speech and as a means of input in their own right. For example, if a user wants to tell a robot to pick up a certain object among many other objects, it can be difficult to indicate the desired object using only speech. However, if the user combines saying "Pick up that object." with a pointing gesture at the target object, this can be easier to process. Hand gestures can also themselves provide strong indications of the users intentions in the absence of speech: for example, users might move their hand near an object in preparation for picking it up, or may hold out their hand to indicate that they need the robot to hand over a particular object.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 308–315, 2008.

In this paper, we introduce and evaluate a method to recognize the following three types of gestures in a human-robot dialog system: pointing, grasping and holding out (Figure 1).

Fig. 1. Pointing, grasping, and holding-out gestures

The paper is organized as follows: we begin by discussing related work in the area of gesture recognition, particularly related approaches to the sub-tasks of image segmentation and classification. In the main part of the paper, we present a fast, robust and easy-to-implement gesture recognition algorithm which differentiates between three gesture classes mentioned above, and which can be extended to other similar applications. After the algorithm has been described in detail, we describe an experiment in which gestures from the JAST project were recognized and classified with an overall accuracy of 92%. At the end of the paper, we draw some conclusions and summarize results of this work.

2 Related Work

Many methods have been developed recently to perform successful gesture recognition. Most of these systems consist of two main steps: segmentation and extraction of invariants, and classification of gestures. In this section we discuss how similar applications perform these two steps.

2.1 Extracting Invariants

Invariants are shape descriptors extracted from an image that are independent of the viewpoint [1]. Using invariants for recognition greatly simplifies the process of object recognition because it allows objects to be compared with reference models regardless of the orientation. Before extracting invariants, it is necessary to segment the recognized image to extract the relevant objects or regions of interest and to omit the irrelevant data.

For hand-gesture recognition, some researchers have tried to do the early segmentation process using skincolor histograms [2,3,4,5]. The problem with these methods is that they do not work well in cases when there are some other objects in the scene with the same color as skin color, or where the hand has other colors. In the target JAST application, the background is static and can easily be eliminated (e.g. using methods described in [6]), so we concentrate instead on the geometric characteristics of the objects.

Zhou *et al.* [2] extracted invariants for gesture recognition using overlapping sub-windows, and characterized them with a local orientation histogram feature description indicating the distance from the canonical orientation. This makes the process relatively robust to noise, but very time-consuming indeed.

Kuno and Shirai [7] used seven invariants to do handgesture recognition, including the position of the fingertip. This is not practical when we have not only pointing gestures, but also several other gestures, like grasping. However, the invariants they extracted inspired us for some future improvements.

2.2 Classification

Classification is a method to assign a class to a point (vector in spaces of more than two dimensions) in N - dimensional space. The classes may be predefined and learned beforehand (supervised learning), or may be extracted automatically based on a similarity metric (unsupervised learning).

A naïve Bayes classifier assigns the most likely class to a given example given its feature vector, simplifying the task greatly by assuming that the features are independent given a class. Such classifiers are robust, simple to implement and computationally efficient – and, despite the often unrealistic assumption of independence, they are frequently very successful in practice. Many techniques have been developed to improve the performance of naïve Base classifiers; Zheng and Webb [8] provide an overview of efforts in this area.

K-nearest neighbors (KNN) classifiers have a good performance when the attributes of a system are linearly separable. It finds the K nearest (already classified) vectors in the space to the input. The class which has the most vectors in those K neighbors is chosen to be the class of the input vector. K-nearest neighbors with distance weighting (KNNDW) is an improvement which has been proved to outperform KNN in many cases [9]. In this method, the contribution of each neighbor to the overall classification is weighted by its distance from the point being classified.

The most relevant work to our method has been performed by Frank *et al.* [10] which introduces a locally weighted naïve Bayes (LWNB) classifier. Their evaluation on UCI dataset shows that LWNB outperforms KNN and KNNDW when K is big enough. Other refinements, like instance cloning local nave Bayes (ICLNB) [11], have also been introduced which manipulate the training data to get a better performance from the Bayes classifier.

In our implementation, we use a combination of KNNDW and LWNB to get a better performance without manipulating the training data or any complicated modification. The complete explanation can be found in Section 4.

3 Preoprocessing

The first step in the gesture-recognition process is to process the raw images from the overhead camera to extract the background and to identify *Regions of Interest* (ROI) for the gesture-recognition process. Once ROIs have been identified, the next step is to extract the geometric invariants from the binary image for use in classification. For each ROI we define the following invariants:

1. Number of points
2. Length of the outer contour
3. Change of gradient in x direction
4. Change of gradient in y direction

Since the user's hands enter the image from outside the camera view, we consider for gesture recognition only those ROIs which end at one of the four sides of the image (table).

For extracting the invariants, only a specific, predefined area of the end part of the ROI is processed. This way a completely stretched hand will be processed from the wrist to the finger tips. Next, the outer contour of the object is extracted, where the length of this contour is the second invariant.

Then we explore the contour to find the x-y gradient changes, which corresponds to the number of changes in direction. We refer to these points as *gradient points*. To avoid noise, we inspect only the changes in direction which last for a known number of steps (three steps in our application).

Supposing that we have m invariants, we have a vector with m (which is four in our application) dimensions.

$$Inv(m) = \{a_1, a_2, \ldots, a_m\} \tag{1}$$

During the training phase (Section 4.1), the resulting vector is added to the training pool; during the classification phase (Section 4.2), it is compared against the three gesture classes for identification.

4 Gesture Recognition

Before performing classification, a training pool is created based on a range of gestures produced by different users, where each training instance is labeled with its gesture class. The invariants from this pool are stored for use in the classification process. In Section 4.1, we describe the training process, while in Section 4.2 we describe how classification proceeds.

Note that we are classifying static gestures, while the user's hands could be in motion. We therefore wait for the system to reach a stable state before performing training or classification. A stable state is detected by tracking the coordinates of the ROIs and initiating gesture recognition only once the coordinates remain constant for several frames.

4.1 Training Phase

For the training phase, the user moves his or her hand in different positions and angles for each of the gestures, using both the left and right hands. As stated before, we have three classes of gestures:

$$C(m) = \{c_1, c_2, c_3\} \tag{2}$$

All the extracted invariants are saved in a simple text file. It is recommended that the training is done with a couple of users with different hand size and shape so that the classifier becomes more robust. In our application we used four users' hands. For each gesture around 150 samples is sufficient, so at the end of the training process we have a file consisting of 500 to 600 labeled gestures.

The vectors in the file have one more dimension in comparison with the invariant vector because of the class-id. If we assume that there are n vectors in the file (n samples) then each vector will be:

$$\begin{gathered} Tr_n(m) = \{i_0, i_1, \ldots, i_m\} \\ i_0 \subseteq C \end{gathered} \tag{3}$$

After constructing this pool of labeled invariant vectors, classification is able to proceed.

4.2 Classification Phase: Combining KNNDW and LWNB

To classify the extracted invariant, we first find the K nearest neighbors which are calculated based on the weighted distance of each training vector to the input invariant. Formally, we define the distance-weighting vector as:

$$W_{dist}(m) = \{wDist_1, wDist_2, \ldots, wDist_m\} \tag{4}$$

The distance from the extracted invariant to training vector n can then be computed in Euclidean space as follows:

$$dist_n(tr, Inv) = \sqrt{\sum_{i=1}^{m} \frac{\left(Tr_n(i) - Inv(i)\right)^2}{wDist_i}} \tag{5}$$

We also normalize the distance so that all the values will be in $[0, 1]$. Next, we choose the K vectors from the training pool which have the shortest distance to the given invariant.

$$\begin{gathered} c(x) = \{Tr_x(1), dist_x\} \\ x = \{1, \ldots, K \end{gathered} \tag{6}$$

A normal naïve Bayes probability for the class of the given invariant will then be

$$p\left(C(j)|x\right) = \frac{\sum_{x=1}^{K} \delta\left(C(j), c(x)\right)}{K} \tag{7}$$

$$\delta(a,b) = \begin{cases} 1 \text{ if } a = b \\ 0 \text{ otherwise} \end{cases} \tag{8}$$

where j is the index of each class (type of gesture).

To improve the result, we add weights to the neighbors. This weight $wB(x) = f(d_x)$ is a function of the Euclidian distance of each vector d_x, which can be any monotonically decreasing function. In our application, we experimented with a functions like $f(d_x) = (1 - d_x)$ or $f(d_x) = (d_x)^{-p}$ for various p, but the function which produces the best classification performance was:

$$wB(x) = f(d_x) = \frac{1 - d_x}{1 + d_x} \tag{9}$$

Using these weights, we define our locally weighted naïve Bayes probability by weighting equation 4.2 as

$$p\left(C(j)|x\right) = \frac{\sum_{x=1}^{K} wB(x)\delta\left(C(j), c(x)\right)}{\sum_{y=1}^{K} wB(y)} \tag{10}$$

Then we can simply choose the class with the highest probability.

$$c(Inv) = \text{argmax}_{j=1,\ldots,3} p\left(C(j)|x\right) \tag{11}$$

The classification algorithm can be summarized as follows:

1. Find the k-nearest neighbors with weighted distance from the training pool.
2. Find the naïve Bayes probability of each, while weighted disproportional to their distance.
3. Choose the class of the vector with the highest probability.

5 Experimental Results

In order to test the recognition algorithm we constructed a training pool with less than 200 samples for each of the gestures, for a total of 580 samples in the training pool. These samples were made by three persons in different lighting conditions. Then we created a testing pool with about 40 samples for each gesture by a person other than those three whose gestures were represented in the training pool.

The highest overall performance without weighting the invariants shows 91.3% correct classifications with $K = 4$.

After trying many combinations of weights on the members of invariants, we found the best weighting vector $wDist$ to be $W_{dist}(m) = \{0.6, 0.6, 1.0, 1.0\}$. That is, the gradient changes are both weighted at 1.0, while the number of points and contour length are weighted at 0.6. This result is intuitively acceptable: the range of objectpoints and length of contour is much wider than the number of changes in gradients.

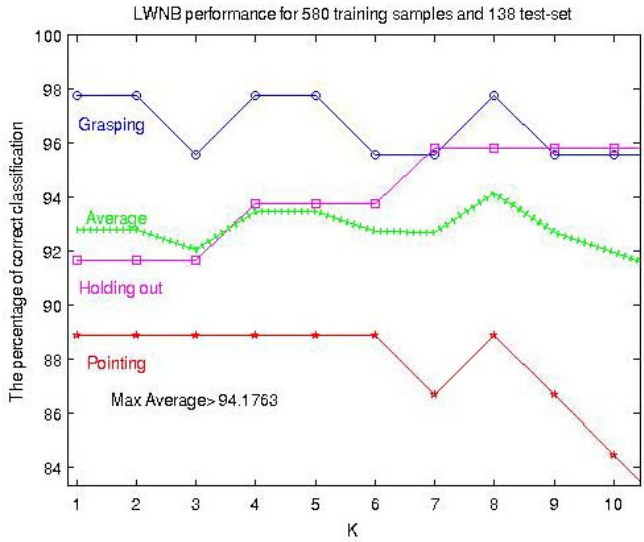


Fig. 2. Classification result with distance weighting

Testing the recognition system with distance weighting, we achieved the results shown in Figure 2. It is quite obvious that after applying distance weighting the performance increases and the fluctuation decreases. We observed that when $K \geq 7$ the performance starts to sink. The best performance seemed to be for $4 \leq K \leq 7$.

Running the full gesture-recognition process on a frame takes less than 50 msec on average. Of this time, segmentation takes 20–30 msec, while the recognition process takes 10–20 msec.

6 Conclusion

We have described a static gesture recognition method to distinguish between three gestures types: pointing, grasping and holding out. The process is based on classifying invariants of image blocks using a locally weighted naïve Bayes and K-nearest neighbors classifier.

The preprocessing is done by an adaptive method first extracting the background, which is considered to be unicolored (surface of the table), and segmenting the remaining pixels into regions of interest afterwards.

Four invariants of each ROI are then extracted. These invariants are: Number of pixels, length of the outercontour and changes in x and y gradients. The extracted invariants are then compared against the invariants in a pool of labeled examples created during a training phase for each type of the gestures. The best suitable type of gesture is then given by using a locally weighted nave Bayes classifier which is fed by the K-nearest neighbors of each invariant in the invariants pool.

After classification, an appropriate algorithm is applied in order to obtain symbolic information according to the type of gesture. This information is the finger-tip and its angle for pointing gestures, area of grasping for grasping gestures and the center of hand-pit for holding out type.

In an experiment, the whole process takes less than 50 msec in total and has an overall performance of about 93% at identifying the correct gesture type.

Acknowledgements

We thank the Cognitive Robotics Group at TUM for useful discussions and suggestions, particularly Thomas Rückstieß and Christian Osendorfer.

This research was supported by the EU project JAST (FP6-003747-IP).

References

1. Weiss, I.: Geometric invariants and object recognition. Int. J. Comput. Vision 10(3), 207–231 (1993)
2. Zhou, H., Lin, D.J., Huang, T.S.: Static hand gesture recognition based on local orientation histogram feature distribution model. In: Proc. IEEE CVPR Workshop, p. 161 (2004)
3. Hongo, H., Ohya, M., Yasumoto, M., Yamamoto, K.: Face and hand gesture recognition for human-computer interaction. In: Proc. IEEE 15th Int. Conf. Pattern Recognition, vol. 2, pp. 921–924 (2000)
4. Wu, H., Shioyama, T., Kobayashi, H.: Spotting recognition of head gestures from color image series. In: Proc. ICPR, Washington, DC, USA, vol. 1, p. 83. IEEE Computer Society, Los Alamitos (1998)
5. Boehme, H., et al.: User localization for visually-based human-machine-interaction. In: Proceedings International Conference on Automatic Face- and Gesture Recognition, Washington, DC, USA, pp. 486–491. IEEE Computer Society, Los Alamitos (1998)
6. McIvor, A.: Background subtraction techniques. In: Proc. of Image and Vision Computing, Auckland, New Zealand (2000)
7. Kuno, K., Shirai, Y.: Manipulative hand gesture recognition using task knowledge for human computer interaction. In: Proc. of International Conference on Face & Gesture Recognition, Washington, DC, USA, p. 468. IEEE Computer Society, Los Alamitos (1998)
8. Zheng, Z., Webb, G.I.: Lazy learning of bayesian rules. Mach. Learn. 41(1), 53–84 (2000)
9. Morin, R.L., Raeside, B.E.: A reappraisal of distance-weighted k-nearest neighbor classification for pattern recognition with missing data. IEEE Transactions on Systems, Man, and Cybernetics SMC-11(3), 241–243 (1981)
10. Frank, E., Hall, M., Pfahringer, B.: Locally weighted naive bayes. In: Proc. of UAI 2003, pp. 249–256. Morgan Kaufmann, San Francisco (2003)
11. Jiang, L., Zhang, H., Su, J.: Learning k-nearest neighbor naive bayes for ranking. In: Li, X., Wang, S., Dong, Z.Y. (eds.) ADMA 2005. LNCS, vol. 3584, pp. 175–185. Springer, Heidelberg (2005)

SBUQA Question Answering System

Mahsa A. Yarmohammadi, Mehrnoush Shamsfard, Mahshid A. Yarmohammadi, and Masoud Rouhizadeh

Natural Language Processing Laboratory, Shahid Beheshti University, Tehran, Iran
m_yarmohammadi@std.sbu.ac.ir, m-shams@sbu.ac.ir,
arabyarm@ce.sharif.edu, m.rouhizadeh@mail.sbu.ac.ir

Abstract. In this paper we propose a model for answer extraction component of a question answering system called Sbuqa. In our proposed system we exploit methods for meaning extension of the question and the candidate answers and also make use of ontology (WordNet). We use LFG -Lexical Functional Grammar, a meaning based grammar that analyses sentences in a deeper level than syntactic parsing- to represent the question and candidate answers. we proposed an algorithm called extended unification of f-structures to match the f-structure pattern of the question and f-structure patterns of candidate answers. Four main levels of matching are defined based on the exact matching, approximate matching, or no matching between slots and fillers of the two f-structure patterns. Finally, the sentences which acquire the minimum score to be offered the user are selected; the answer clause is identified in them and displayed to the user in descending order.

Keywords: Question answering systems, Answer extraction,Lexical Functional Grammar, wh-question, Information retrieval, Natural language processing.

1 Introduction

Methods which extract answers based on only the keywords ignore many acceptable answers of the question. Therefore, in our proposed system we approach to methods for meaning extension of the question and the candidate answers and also make use of ontology (WordNet). In order to match the question and the candidate answers we use Lexical Functional Grammar and the benefits of its functional-structure representation, and propose a unification algorithm. The question answering system we have designed and implemented for this purpose is named SBUQA[1].

In the following sections, we first introduce the overall architecture of QA systems, Lexical Functional Grammar and advantages of using this grammar in QA systems. Then, we present SBUQA. Finally, we describe the evaluation of SBUQA and mention future works.

2 Lexical Functional Grammar

Lexical Functional Grammar (LFG) [7] is a meaning based grammar. In this formalism the sentences are analyzed at a deeper semantic level than only syntactic parsing

[1] Shahid Beheshti University Question Answering system.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 316–323, 2008.

[1]. This type of analysis is useful in that it is a more abstract representation of linguistic information than a parse tree structure. In addition, long distance dependencies, which are very common in interrogative sentences and fact seeking questions, are resolved in order to have a complete and correct f-structure analysis. This makes LFG analysis useful for QA tasks because it identifies the focus of the question and also the functional role that the focus can fulfill. Another advantage of LFG on parse tree grammas is its language-independence. LFG f-structure are same among different languages, so it is possible to have a multilingual question answering system.

LFG f-structure is used in SBUQA in the process of representing and matching the question and its candidate answers.

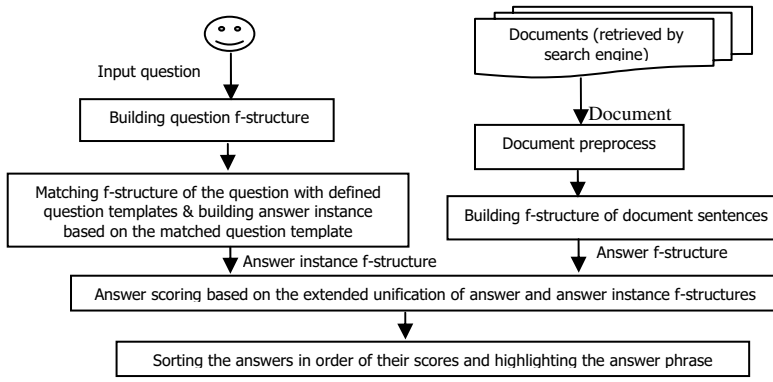


Fig. 1. The architecture of SBUQA

3 SBUQA System

From overall view, the third component of a QA system, gets the user question and a set of retrieved text documents as the input, and shows user the answer(s) extracted from the document set as the output. The document set is composed of text documents that are the output of the second component of the QA system (search engine). Figure 1 shows the architecture of SBUQA.

The components of the system and relationships between them are described in the following.

3.1 Getting Question and Building Its f-Structure

The system gets the user natural language question and sends it to the LFG parser to build its f-structure representation. This representation makes one of the inputs of "Building f-structures of Answer Instances" component.

3.2 Document Preprocessing and Building Its Sentences f-Structures

Documents retrieved by the search engine are saved as text files in the system's document bank. These documents are preprocessed by JAVARAP[2] tool, so that the

[2] http://www.comp.nus.edu.sg/~qiul/NLPTools/JavaRAP.html

sentences are separated and pronouns are replaced by their referents. Then, the sentences of each document are sent to the LFG parser to represent as f-structures (called fs_C).These representations make one of the inputs of "Extended Unification Algorithm and Answer Scoring" component.

3.3 Building f-Structures of Answer Instances

We have defined some templates -represented in f-structure format- for Wh questions (called fs_{TQ}). Question f-structure is compared to fs_{TQ}s and is matched with one of them, say X. For each fs_{TQ}, we have defined one or more answer template(s) represented in f-structure format (called fs_{TA}). fs_{TA}s of the X (the fs_{TQ} matching with user question) are filled with question keywords and make answer instances (called fs_A). These instances are the other inputs of "Scoring the answer" component. Question and answer templates are described in section 3.5.

3.4 Extended Unification Algorithm and Answer Scoring

fs_C of each sentence of document (input from document preprocessing component) is compared to fs_As (input from "Building f-structures of Answer Instances" component). The comparison is done by the Extended Unification Algorithm, introduced in section 3.5 and sentences are scored. Finally, sentences acquiring the score more than a defined threshold are selected, ordered by their scores, and shown to the user.

3.5 Question and Answer Templates

We define templates for questions and related answers based on the following categorization of English sentences [2]:

Active sentences with transitive verb, containing subject, verb and optional object.

Active sentences with intransitive verb, containing subject and verb.

Passive sentences, containing subject (promoted object of the active form) verb, and an optional by-phrase containing object (demoted subject of the active form).

Sentences containing copula.

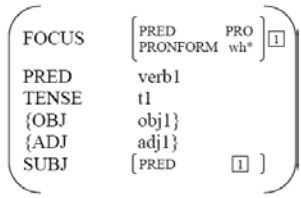
Each of the above sentences can contain some complement or adverb.

We tried to define question templates for wh questions so that they can cover all standard forms of questions. We considered five types of wh questions: who, where, when, what and which. Regarding four forms described above, we defined the following templates for wh questions.

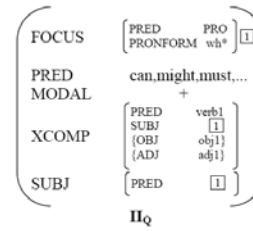
3.5.1 Wh-Questions

The following four templates are the question templates of forms 1 and 2 (active sentences). These templates are numbered from I to IV. The FOCUS property indicates the type of wh question. The PRED property indicates the main verb of the sentence, TENSE indicates the tense of the verb, OBJ represents the object, ADJ represents the adjuncts especially adverb, SUBJ indicated the subject and XCOMP represented the complement The MODAL property in template II indicates that the sentence contains a modal verb.

Template I is used for active interrogatives that contains only the main verb and template II covers active interrogatives that contain modal verb in addition to main verb.

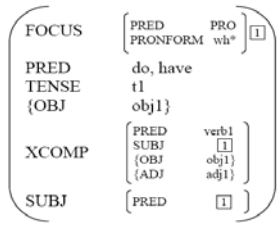


I_Q

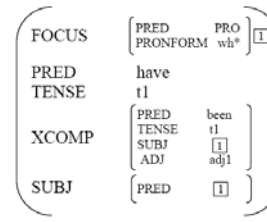


II_Q

Templates III and IV covers interrogatives that use auxiliary verb do or have.

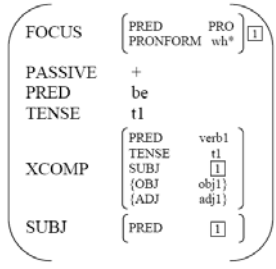


III_Q

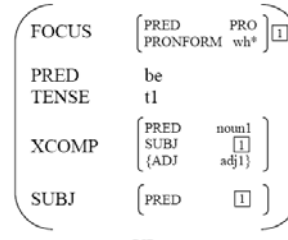


IV_Q

Template for passive interrogatives (form 3) is as template V. The PASSIVE property with + represents the passive sentence. Template for copula interrogatives (form 4) is as VI.



V_Q

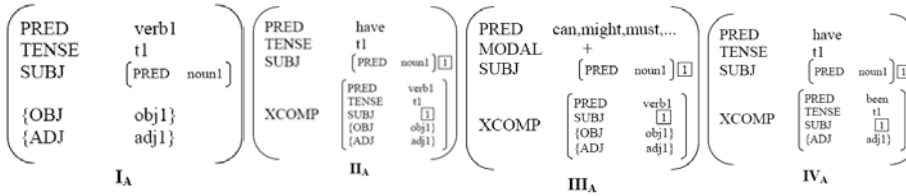


VI_Q

For each of the defined question templates (fs_{TQ}), one or more answer templates (fs_{TA}) are defined. As mentioned in section 3-2-3, if the user question matches with one of the fs_{TQ}s, the fs_{TA}s for that are filled with words of the question in order to make answer instances and are used in the extended unification algorithm with sentences of candidate answer.

Template for answer of active interrogatives – that matches with forms 1 and 2- are as the followings:

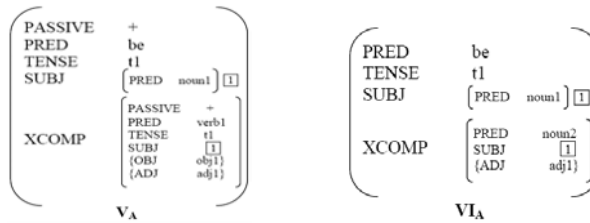
Answer template I_A is defined based on question template I_Q. As the same, the answer template II_A is defined for question template II_Q, III_A for III_Q and IV_A for IV_Q.



A question in active form can be answered by a passive sentence; so the template for passive answer (V_A) is added to answer templates for form 1 and 2. Answer templates for questions that match form 3 (passive sentences) are as follows:

Answer template V_A is defined based on question template V_Q. Here it is possible that the question in passive form have answer in active form. Hence four answer templates for active sentences (I_A, II_A, III_A, IV_A) are also added to answer templates of form 3.

Answer template of the questions matched with form 4 (copula) is like the following. This template is defined based on the question template VI_Q



3.6 Extended f-Structure Unification

Answer extraction is a result of unifying the f-structure of the candidate answer and instance of the answer (that is generated based on the question). Experiments Shows that unification strategy based on exact matching of values is not sufficiently flexible [5]. For example, sentence "Benjamin killed Jefferson." is not answer to question "Who murdered Jefferson?" by exact matching. In our proposed system, we considered approximate and semantic matching in addition to exact and keyword-based matching. Approximate matching is performed by ontology based extended comparison between different parts of the question template and the candidate answer template (including subj, obj, adjunct, verb and …) and comparing of their types.

In Our unification algorithm, by slot we mean various parts of templates (including subj, obj, adjunct, verb and …) accompanied with their types, and by filler we mean values (instances) that slots are filled with them.

For determining the level of matching between fs_A and fs_C, we proposed a hierarchical pattern based on the exact matching, approximate matching, or no matching between slots and fillers of the two structures. Levels of scoring the candidate answer abased on the matching of fs_A and fs_C is as follows:

A) Existence of all slots of fs_A in fs_C and
   Exact matching of the fillers.
   Approximate matching of the fillers.
   No matching of the fillers.

B) Existence of all slots of fs_A in fs_C, plus additional slots in fs_C and
Exact matching of the fillers.
Approximate matching of the fillers.
No matching of the fillers.
C) Existence of some (no all) slots of fs_A in fs_C and
Exact matching of the fillers.
Approximate matching of the fillers.
No matching of the fillers.
D) Existence of some slots of fs_A in fs_C, plus additional slots in fs_C and
Exact matching of the fillers.
Approximate matching of the fillers.
No matching of the fillers.

For approximate matching of the fillers, a hierarchical pattern as the following is defined:

Approximate matching of the fillers of type verb:
-Value of fs_C is synonym of value of fs_A.
-Value of fs_C is troponym of value of fs_A.
-Value of fs_A is troponym of value of fs_C.
-Value of fs_A is hypernym of value of fs_C.
-Value of fs_C is hypernym of value of fs_A.

Approximate matching of the fillers of other parts of the sentence (obj, subj, adjunct):
-Value of fs_C is synonym of value of fs_A.
-Value of fs_A is hypernym of value of fs_C.
-Value of fs_C is hypernym of value of fs_A.
-Value of fs_C is meronym of value of fs_A.
-Value of fs_C is holonym of value of fs_A.

4 SBUQA Evaluation

For evaluating the operation of the proposed system in finding final answer, we selected 10 questions (two questions of each type of where, who, what, when and which) from TREC question set. These questions selected in a way that cover various kinds of question templates. For each question, sentences of documents retrieved by google search engine and AnswerBus, Start and Ask online question-answering systems are extracted. A total number of 100 sentences are retrieved and are used in evaluation. Level of matching of these sentences with answer templates, are determined using the implemented tool. If the sentence matches with one of the templates, the answer part is extracted from the sentence using the tool and correctness or incorrectness of it is determined.

Number of matched sentences with one of the answer patterns (A, B, C, or D scoring levels and exact, approximate, or no matching types) are given in Table 1. Column *T* shows the number of sentences that contain the correct answer and are recognized by one of the answer patterns correctly. Column *F* shows the number of sentences matched with on of the patterns but contain an incorrect answer and are recognized incorrectly by the system. Column *nA* (not Answer) is for sentences

matched with one of the patterns but the system could not extract any answer. It happens when a filler of the same type of expected answer type is not recognized in the sentence by system. If the sentence does not have such a filler, value of nA_T column is increased but if the sentence does have such a filler and the system did not recognize it by mistake, value of nA_F column is increased. Among 100 sentences, 76 sentences matched with one of patterns. 23 other sentences do contain the answer but did not recognized by any of the patterns, these are shown by nE (not Extracted) parameter.

Table 1. Number of matched sentences with onr of answer patterns (among 100 sentences)

Matching Type	A				B				C				D			
	T	F	nA		T	F	nA		T	F	nA		T	F	nA	
			T	F			T	F			T	F			T	F
Exact	10	0	5	0	4	2	0	0	2	2	0	0	1	1	0	0
Approximate	8	5	4	1	2	1	0	0	1	1	0	0	0	1	0	0
No matching	0	7	3	0	1	1	0	0	0	1	0	0	1	2	0	0

Based on the results of Table 1, exact and approximate matchings in all the 4 scoring levels extract the answer in an acceptable reliability, but no-matching situation is not reliable and often offers an incorrect answer. Precision of matching type i (A, B, C, and D) is calculated by the following formula:

$$P_i = \frac{\text{correct extracted answers}}{\text{all extracted answers}} = \frac{T_i}{T_i + F_i} \quad (1)$$

Recall of the system is calculated by the following formula:

$$r = \frac{\text{correct extracted answers}}{\text{all correct answers}} = \frac{T}{T + nA_F + nE} \quad (2)$$

Based on the above formulas, the precision of matching level A is equal to 0.78, level B is equal to 0.67, level C is equal to 0.50, and the precision of level D is equal to 0.33. The recall of the system is equal to 0.54.

Also, each test question is evaluated based on some QA systems metrics: First Hit Success (FHS), First Answer Reciprocal Rank (FARR), First Answer Reciprocal Word Rank (FARWR), Total Reciprocal Rank (TRR), and Total Reciprocal Word Rank (TRWR). Possible values for FHS, FARR, FARWR, and TRWR are in the range of 0 to 1 and the ideal value in an errorless QA system is equal to 1. Possible values for TRR are from 0 to ∞ (it doesn't have an upper bound). The greater value for TRR, indicates that more correct answers are extracted. System evaluation based on these metrics, gives 0.78 for FHS and FARWR, 0.82 for FARR, 1.33 for TRR, and 0.72 for TRWR, that are good (greater that the average) values.

5 Conclusion and Future Works

The SBUQA system that is proposed for the third component of question-answering systems, operates based on f-structure of the question and candidate answers and

extended unification based on ontology (WordNet). According to the evaluation measures of question-answering systems, the SBUQA system resulted a good (better that average) operation in retrieving final answer.

The proposed system is designed for wh questions in open domain. Further extensions can cover yes/no questions and other types of questions. f-structure is beyond some shallow representations that are dependent to language. Although languages are different in shallow representations, but they can be represented by the same (or very similar) syntactic (and semantic) slot-value structures. This feature of f-structure makes it possible to use the algorithms introduced in the proposed system for other languages including Persian. Now it is not possible to implement the system for Persian, because lack of usable and available tools for processing Persian language (such as parser and WordNet ontology for Persian). But we consider this as future extensions of the system.

References

1. Yarmohamadi, M.A.: Organization and Retrieving Web Information Using Automatic Conceptualization and Annotation of Web Pages, MS dissertation, Computer Engineering Department, Faculty of Engineering, Tarbiat Modarres University, Tehran, Iran (2006)
2. Dabir-Moghadam, M.: Theoritical Linguistics:Emergence and Development of Generative Grammar, 2nd edn. Samt Publication, Tehran (2004)
3. Eshragh, F., Sarabi, Z.: Question Answering Systems. BS dissertation, Electrical & Computer Engineering Department, Shahid Beheshti University, Tehran, Iran (2006)
4. Judge, J., Guo, Y., Jones, G.J.: An Analysis of Question Processing of English and Chinese for the NTCIR 5 Cross-Language Question Answering Task. In: Proceedings of NTCIR-5 Workshop Meeting, Tokyo, Japan (2005)
5. Lin, J.J., Katz, B.: Question answering from the web using knowledge annotation and knowledge mining techniques. In: Proceedings of the ACM Int. Conf. on Information and Knowledge Management, CIKM (2003)
6. Von-Wun, S., Hsiang-Yuan, Y., Shis-Neng, L., Wen-Ching, C.: Ontology-based knowledge extraction from semantic annotated biological literature. In: The Ninth Conference on Artificial Intelligence and Applications (2004)
7. Molla, D., Van Zaanen, M.: AnswerFinder at TREC 2005. In: Proceedings of the Fourteenth Text REtrieval Conference Proceedings (TREC 2005), Gaithersburg, Maryland, The United States (2005)
8. Kil, J.H., Lloyd, L., Skiena, S.: Question Answering with Lydia. In: Proceedings of the Fourteenth Text REtrieval Conference Proceedings (TREC 2005), Gaithersburg, Maryland, The United States (2005)

A New Feedback ANC System Approach

Pooya Davari and Hamid Hassanpour

Department of Electrical and Computer Engineering,
Mazandaran University, Babol, Iran
pdavari@stu.nit.ac.ir, h.hassanpour@nit.ac.ir

Abstract. We propose a new active noise control (ANC) technique. The technique has a feedback structure to have a simple configuration in practical implementation. In this approach, the secondary path is modelled online to ensure convergence of the system as the secondary paths are practically time varying or non-linear. The proposed method consists of two steps: a noise controller which is based on a modified FxLMS algorithm, and a new variable step size (VSS) LMS algorithm which is used to adapt the modelling filter with the secondary path. The proposed algorithm stops injection of the white noise at the optimum point and reactivate the injection during the operation, if needed, to maintain performance of the system. Eliminating continuous injection of the white noise increases the performance of the proposed method significantly and makes it more desirable for practical ANC systems. The computer simulations are presented to show the effectiveness of the proposed method.

Keywords: Active Noise Control, Adaptive Filter, FxLMS, Feedback, Secondary Path, Online Modelling.

1 Introduction

Active noise control (ANC) is an electro acoustic system that efficiently attenuates low frequencies unwanted noises (primary noise) where passive methods are either ineffective or tend to be very expensive or bulky. An anti-noise of equal amplitude and opposite phase replica primary noise is tried to be generated and then combined with the unwanted disturbance. Following the superposition principle the result is cancellation or reduction of both noises [1].

The secondary path is practically nonlinear and introduces delay. These characteristics cause instability problem to the standard Least Mean Square (LMS) algorithm, resolving the instability problem requires using FxLMS algorithm [1,2]. The FxLMS algorithm uses estimation of the secondary path to overcome the problem raised by the above-mentioned characteristics of the secondary path.

Estimation of the secondary path is performed offline prior to the implementation of ANC algorithm, but in practical cases the secondary path are usually time varying or non-linear. Consequently, online modelling of secondary path is required to ensure the convergence of the ANC algorithm [3].

The proposed system is based on modified versions of FxLMS and variable step size (VSS) LMS algorithm [4]. Here we adapt the FxLMS and VSS-LMS algorithms with reference signal power variations.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 324–331, 2008.

In this research, we use the advantage of using random noise in online secondary path modeling for a feedback ANC system. To increase performance of the algorithm we stop the VSS-LMS algorithm at the optimum point. This means stopping the injection of the white noise. Not continually injection of the white noise makes the system more desirable especially in ANC headphones applications.

The rest of the paper is organized as follows. Section 2 introduces our proposed method. In Section 3 we illustrate our simulation results, and finally in Section four conclusions are drawn.

2 Proposed Method

As mentioned before, the secondary path is practically nonlinear and introduces delay. Hence, the secondary path is preferably modelled online to overcome on these characteristics.

Fig. 1 shows block diagram of the proposed system. The secondary signal *y(n)* is expressed as:

$$y(n) = \mathbf{w}^T(n)\mathbf{x}_L(n). \tag{1}$$

The ordinary FxLMS algorithm [1] updates the control filter *W(z)* as below:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu_w(n) f(n) \hat{\mathbf{x}}'(n), \tag{2}$$

$$f(n) = [d(n) - y'(n) + v'(n)] - \hat{v}'(n), \tag{3}$$

where *d(n)* is the primary noise, $y'(n) = s(n) * y(n)$ is the secondary cancelling signal, $v'(n) = s(n) * v(n)$ is the filtered training signal, *f(n)* is the error signal, and μ_w is the step size parameter.

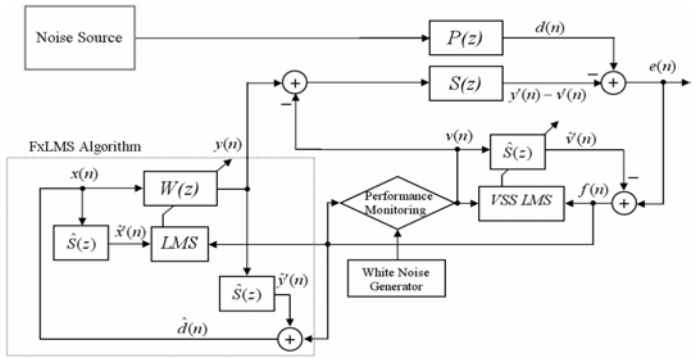


Fig. 1. Block diagram of the proposed feedback ANC system

Here $\hat{x}'(n)$ is the filtered-reference signal which is given as:

$$\hat{x}'(n) = \hat{\mathbf{s}}^T(n)\mathbf{x}_M(n), \tag{4}$$

where $\hat{\mathbf{s}}(n) = [\hat{s}_0(n)\hat{s}_1(n)...\hat{s}_{M-1}(n)]^T$ and $\mathbf{x}_M(n) = [x(n), x(n-1), ..., x(n-M+1)]^T$.

Here we suggest a new version of the FxLMS algorithm to increase noise attenuation. Usually μ_w in (2) is set to a low value. This prevents the system to diverge when power of the reference signal *x(n)* is increased. However, once the power decreases the low value of μ_w reduces the noise attenuation and convergence rate of the adaptive filter (*W(z)*). Thus, if μ_w could be increased when the power decreases, and vice versa, the system performance would be risen significantly. Thereby we modified (2) as follows:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{1}{\sqrt{P_{x-e}(n)}} \mu_w(n) f(n) \hat{\mathbf{x}}'(n), \tag{5}$$

where $\hat{\mathbf{x}}'(n) = [\hat{x}'(n), \hat{x}'(n-1), ..., \hat{x}'(n-L+1)]^T$ is the signal vectors of length *L*, where *L* is the order of the filter *W(z)*. Here $P_{x\text{-}e}(n)$ is given as:

$$P_{x-e}(n) = \gamma P_{x-e}(n-1) + (1-\gamma)(x(n) - e(n))^2, \; 0.9 < \gamma < 1. \tag{6}$$

The residual error signal *e(n)* of this algorithm is expressed as:

$$e(n) = d(n) - y'(n) + v'(n) \tag{7}$$

where *v(n)* is an internally generated white Gaussian noise, which is injected at the output of the control filter *W(z)*.

As we know $\hat{S}(z)$ is the modelling FIR filter with length *M* that generates $\hat{v}'(n)$ as expressed below:

$$\hat{v}'(n) = \hat{\mathbf{s}}^T(n) \mathbf{v}_M(n) \tag{8}$$

where $\mathbf{v}_M(n) = [v(n), v(n-1), ..., v(n-M+1)]^T$ and $\hat{\mathbf{s}}(n) = [\hat{s}_0(n) \hat{s}_1(n) .. \hat{s}_{M-1}(n)]^T$, *M* is order of the $\hat{S}(z)$. Coefficients of the modelling filter $\hat{S}(z)$ in VSS-LMS algorithm [4] are updated as follows:

$$\hat{\mathbf{s}}(n+1) = \hat{\mathbf{s}}(n) + \mu_s(n) f(n) \mathbf{v}(n), \tag{9}$$

where $\mathbf{v}(n) = [v(n), v(n-1), ..., v(n-M+1)]^T$ and $\mu_s(n)$ is the step-size parameter of the VSS-LMS algorithm which will be explained later.

We add the above new term to (9) to maintain μ_s variations with μ_w as follows:

$$\hat{\mathbf{s}}(n+1) = \hat{\mathbf{s}}(n) + \frac{1}{\sqrt{P_{x-e}(n)}} \mu_s(n) f(n) \mathbf{v}(n). \tag{10}$$

The output signal $\hat{y}'(n)$ is used to define estimation of the primary noise $\hat{d}(n)$:

$$\hat{d}(n) = f(n) + \hat{y}'(n) \tag{11}$$

where $x(n) \equiv \hat{d}(n)$ is our reference signal, and $\hat{y}'(n)$ is expressed as follow:

$$\hat{y}'(n) = \hat{\mathbf{s}}^T(n)\mathbf{y}_M(n) \tag{12}$$

The VSS-LMS algorithm introduced in [4] is used to update modelling filter $\hat{S}(z)$ coefficients. For more detail on theory of this algorithm reader may refer to [4, 5]. As we mentioned before, the modelling filter in equation (9) and (10) is updated using the step-size parameter ($\mu_s(n)$) of VSS-LMS algorithm and this parameter is calculated using the following three steps [4]:
Initially, the power of error signals *e(n)* and *f(n)* are computed:

$$\begin{aligned} P_e(n) &= \lambda P_e(n-1) + (1-\lambda)e^2(n) \\ P_f(n) &= \lambda P_f(n-1) + (1-\lambda) f^2(n), \ 0.9 < \lambda < 1. \end{aligned} \tag{13}$$

Then, the ratio of the estimated powers is obtained:

$$\begin{aligned} \rho(n) &= P_f(n)/P_e(n) \quad . \\ \rho(0) &\approx 1, \ \lim_{n\to\infty} \rho(n) \to 0 \end{aligned} \tag{14}$$

The Finally, the step size is calculated as follows:

$$\mu_s(n) = \rho(n)\mu_{s_{\min}} + (1-\rho(n))\mu_{s_{\max}}, \tag{15}$$

where $\mu_{s_{\min}}, \mu_{s_{\max}}$ and λ are the experimentally determined values. These values are selected so that the adaptation is neither too slow nor it becomes unstable.

The main point which results in an increased noise attenuation and convergence rate is due to preventing continuous injection of white noise during system operation. Thereby, the modelling algorithm must be stopped at the point where the modelling filter accuracy is sufficiently high, called the optimum point.

Here, the VSS-LMS algorithm is briefly described to show the way the optimum point is obtained. During the process of this algorithm, μ_s is increased as the error signal *f(n)* decreases and vice versa. Hence, the modeling filter, $\hat{S}(z)$, converges to a good estimation when *f(n)* decreases. This happens when μ_s increases as high as $\mu_{s_{\max}}$. Thus, the injection of the white noise is stopped at the optimum point which is measured using:

$$\mu_{s_{\max}} - \mu_s < \alpha \quad , \ 1\times10^{-4} < \alpha \le 1\times10^{-2}. \tag{16}$$

As the number of iteration increases, equation (16) gets closer to zero. In this equation α is a parameter obtained experimentally. At this point, $\hat{S}(z)$ converges to a good estimation of $S(z)$. As can see from Fig. 1, this condition validity is monitored at the performance monitoring stage. In some practical cases the secondary path may suddenly change. This event derives system to diverge. To prevent this effect we have to update $\hat{S}(z)$.

The proposed algorithm is design in the way that it monitors the secondary path changes by the following expression:

$$20\log_{10}\left|f(n)\right| < 0\,. \tag{17}$$

If the validity of the above equation does not satisfy, the system reactivates the VSS-LMS algorithm and injects white noise to remodel $\hat{S}(z)$. The same as before, the injection is stopped at the optimum point using (16).

3 Simulations

In this section the proposed ANC system is simulated using Matlab version 7.1. In this simulation, we have used the primary path *P(z)* and secondary path *S(z)* of the experimental data provided in [6]. Using these data, *P(z)* and *S(z)* are considered as FIR filters with tap–weight lengths 48 and 16 respectively. Rate of the sampling frequency in this simulation was *2 KHz*. The magnitude responses of these paths are shown in Fig. 2. Length of the filter $\hat{S}(z)$ for modelling the secondary path and length of the adaptive filter *W(z)* used for the noise cancellation have been chosen *16* and *32*, respectively.

To evaluate the performance of the proposed system, we could not find any feedback ANC system with online secondary modelling method to compare with. Thus we implement the existing feedforward ANC systems [4,7] on feedback structure. Except the Eriksson's method which has been presented on feedback structure in [8].

In this simulations performance of the proposed method is compared with that of Akhtar's [4] and Eriksson's method [7]. Extensive experiments have been performed to find suitable values for a fast and stable performance of the ANC system. Simulations parameters for all of the four methods are set for the most proper situation as described in Table 1.

Table 1. Simulation parameters for the three approaches

Akhtar's method $(\mu_w, \mu_{s_{\max}}, \mu_{s_{\min}}, \lambda)$	$7\times10^{-5}, 25\times10^{-3}, 75\times10^{-4}, 0.99$
Eriksson's method (μ_w, μ_s)	$7\times10^{-5}, 1\times10^{-2}$
Proposed method $(\mu_w, \mu_{s_{\max}}, \mu_{s_{\min}}, \lambda, \gamma, \alpha)$	$3\times10^{-4}, 4.5\times10^{-2}, 9\times10^{-3}, 0.99, 0.999, 1.91\times10^{-3}$

We have performed simulations for two separate cases. Comparative results of modelling accuracy and noise reduction for the system are illustrated in Case1. Finally, Case2 indicates effectiveness of the proposed algorithm in maintaining its performance against sudden changes of the secondary path behaviour. In these Cases, the original noise is considered to have several narrowband periodic components, which is usual in ANC applications like those produced by engines, compressors and fans

[9]. It is important to be noted that a white noise with SNR of *30 dB* is added to all of the reference noises used in these Cases.

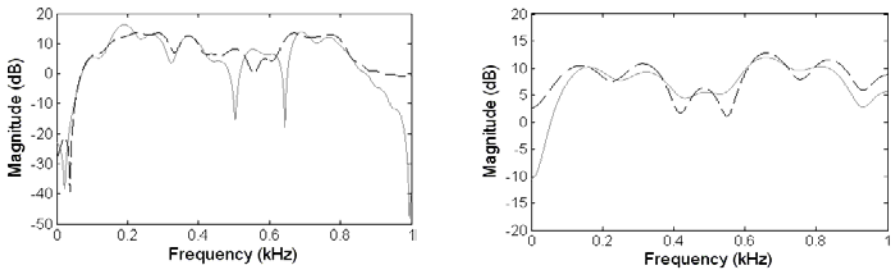
To show the convergence rate and modelling accuracy of the system the relative modelling error is used as defined below:

$$\Delta S(dB) = 10\log_{10}\left\{\frac{\sum_{i=0}^{M-1}[s_i(n)-\hat{s}_i(n)]^2}{\sum_{i=0}^{M-1}[s_i(n)]^2}\right\} \tag{18}$$

To signify performance of the system on noise reduction the following equation is used:

$$R = -10\log_{10}\left(\sum e^2(n) \Big/ \sum d^2(n)\right) \tag{19}$$

All the results shown in each case have been obtained as an average 10 different experiments. To set the initial value for $\hat{S}(z)$ ($\hat{s}(0)$), off-line secondary path modeling is performed. The off-line modeling is stopped when the modeling error (18) is reduced to *-5 dB*.



(a) (b)

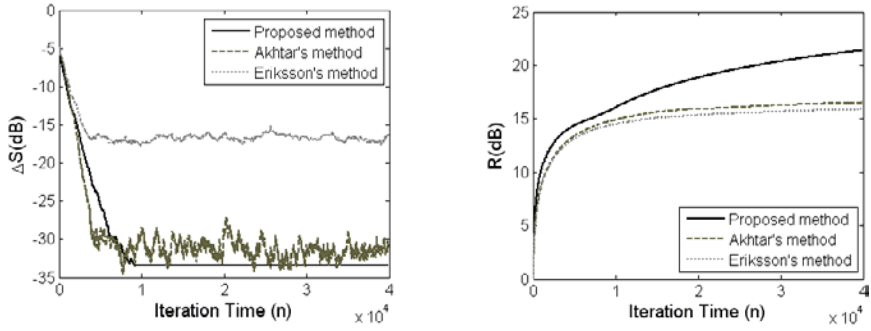
Fig. 2. (a) Magnitude response of the primary path *P(z)*, (b) Magnitude response of the secondary path *S(z)*. (Solid line: original path, dashed line: the changed path at *n=20,000*).

3.1 Case1

Here we compare the performance of the proposed method with the other approaches on the basis of the relative modelling error (18) and noise reduction (19). In this case reference noise is an engine noise at *3700 rpm*. Fig. 3b shows that once injection of the white noise is stopped at the optimum point for the proposed algorithm, the noise reduction ratio is accelerated positively.

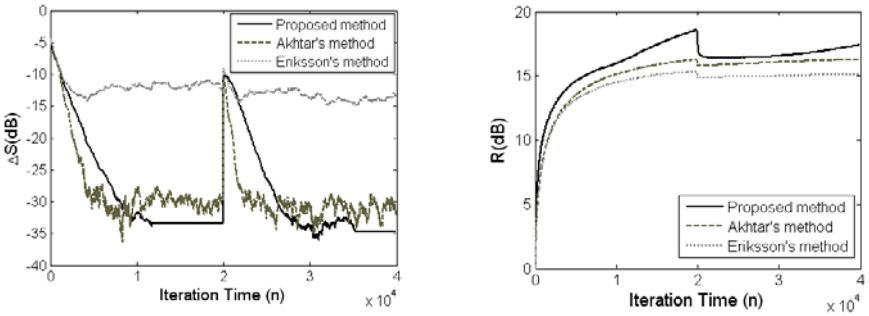
3.2 Case2

In this case we assume that the secondary path transfer function suddenly changes during the operation. Figure 2 shows the magnitude response of the original and changed secondary path. In this figure, the solid line represents secondary path at the



(a) (b)

Fig. 3. Comparison results for the proposed method in Case1 with the other existing approaches. (a) Relative modeling error versus iteration time (*n*), (b) Noise reduction versus iteration time (*n*).



(a) (b)

Fig. 4. Comparison results for the proposed method in Case2 with the other existing approaches. (a) Relative modeling error versus iteration time (*n*), (b) Noise reduction versus iteration time (*n*).

start point, *n = 0*, and the dashed line represents the changed path at iteration *n = 20000*. Here the reference signal reference noise is a narrowband signal comprising frequencies of *0.1-0.9 KHz* with the step of *100 Hz* and Its variance is adjusted to *2*. Figure 4 shows the curve on the basis of the relative modelling error and noise reduction.

The high noise cancellation results in these experiments indicate the accuracy of the proposed system.

4 Conclusions

This paper proposed a feedback ANC system based on a new method for online secondary path modelling. Computer simulations have been conducted for a single-channel feedback ANC system. Simulation results demonstrated that the proposed

method achieved a high performance in noise attenuation. These results indicate efficiency of the secondary path estimation. Preventing continuous injection of the white noise increases the performance of the proposed method significantly and makes it more desirable for practical ANC systems.

References

1. Kuo, S.M., Morgan, D.R.: Active noise control: a tutorial review. Proc. IEEE 8(6), 943–973 (1999)
2. Design of Active Noise Control Systems with the TMS320 Family: Application Report, literature number SPRA042, Texas Instruments (1996)
3. Kuo, S.M., Vijayan, D.: A Secondary Path Modeling Technique for Active Noise Control Systems. IEEE Trans. on Speech and Audio Processing 5(4), 374–377 (1997)
4. Akhtar, M.T., Abe, M., Kawamata, M.: A Method for Online Secondary Path Modeling in Active Noise Control Systems. Proc. IEEE 2005 Intern. Mid. Symp. Circuits Systems 267, I-264–I-267 (2005)
5. Akhtar, M.T., Abe, M., Kawamata, M.: Modified-filtered-x LMS algorithm based active noise control system with improved online secondarypath modeling. In: Proc. IEEE 2004 Intern. Mid. Symp. Circuits Systems, pp. I-13–I-16 (2004)
6. Kuo, S.M., Morgan, D.R.: Active Noise Control Systems-Algorithms and DSP Implementation. Wiley, New York (1996)
7. Eriksson, L.J., Allie, M.C.: Use of random noise for on-line transducer modeling in an adaptive active attenuation system. J. Acoust. Soc. Am. 85(2), 797–802 (1989)
8. Gan, W.S., Mitra, S., Kuo, S.M.: Adaptive Feedback Active Noise Control Headsets: Implementation, Evaluation and its Extensions. IEEE Trans. on Consumer Electronics 5(3), 975–982 (2005)
9. Hu, A.Q., Hu, X., Cheng, S.: A robust secondary path modeling technique for narrowband active noise control systems. In: Proc. IEEE Conf. On Neural Networks and Signal Processing, vol. 1, pp. 818–821 (2003)

Benefiting White Noise in Developing Feedforward Active Noise Control Systems

Pooya Davari and Hamid Hassanpour

Department of Electrical and Computer Engineering, Mazandaran University, Shariatee Av,Babol, Iran
pdavari@stu.nit.ac.ir, h.hassanpour@nit.ac.ir

Abstract. In many applications of active noise control (ANC), an online secondary path modelling method using a white noise as a training signal is required to ensure convergence of the system. The modelling accuracy and the convergence rate increase when a white noise with larger variance is used, however larger the variance increases the residual noise, which decreases performance of the system. The proposed algorithm uses the advantages of the white noise with larger variance to model the secondary path, but the injection is stopped at the optimum point to increase performance of the system. In this approach, instead of continuous injection of the white noise, a sudden change in secondary path during the operation makes the algorithm to reactivate injection of the white noise to adjust the secondary path estimation. Comparative simulation results shown in this paper indicate effectiveness of the proposed method.

Keywords: Active noise control, feedforward, FxLMS, online secondary path modelling.

1 Introduction

Active noise control (ANC) is a technique that efficiently attenuates low frequencies unwanted noises where passive methods are either ineffective or tend to be very expensive or bulky. An ANC system is based on a destructive interference of an anti-noise, which have equal amplitude and opposite phase replica primary noise, with unwanted noise (primary noise). Following the superposition principle, the result is cancellation or reduction of both noises [1].

In ANC systems, the effect shown by the secondary path transfer function, the path leading from the noise controller output to the error sensor measuring the residual noise, generally causes instability to the standard least mean square (LMS) algorithm. In general FxLMS algorithm [1] is used to overcome the instability problem. The FxLMS algorithm uses estimation of the secondary path to reimburse the problem raised by the transfer function. In practical cases the secondary paths are usually time varying or non-linear, which leads to a poor performance or system instability. Therefore, online modelling of secondary path is required to ensure convergence of the ANC algorithm [2, 3, 4, 5].

Most of online secondary path modelling techniques entail injection of the White noise as an ideal training signal in modelling the secondary path [3, 4, 5]. White noise

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 332–339, 2008.

with larger variance improves modelling accuracy and convergence rate [5], nevertheless, with the cost of an increased residual noise. Thus, existing online secondary path modelling techniques use white noise with a low variance to model the secondary path in order to sustain a low residual noise in steady state.

The proposed system is based on an FxLMS algorithm as a main part of the system, and a variable step size (VSS) LMS algorithm [4] is used to adapt the modelling filter of the secondary path. The proposed algorithm uses white noise with large amplitude in modelling the secondary path. However, to increase performance of the algorithm and to prevent the instability effect raised by the large variance white noise, we stop the VSS-LMS algorithm at the optimum point. Stopping at the optimum point increases noise attenuation and allows benefiting advantages of white noise with a larger variance. Not having the white noise makes the system more desirable as continuous existence of the noise in the environment may have an unpleasant result.

A sudden change in the secondary path leads to divergence of the online secondary path modelling filter. To abate this problem, the proposed algorithm is designed in the way that it controls the secondary path changes.

In the existing methods estimation of the secondary path is usually performed offline, prior online modelling, where in the proposed system there is no need of using the offline estimation [3, 4].

Considering the above features in the proposed method assists obtaining a better convergence rate and modelling accuracy, which results in a robust system. We show that this online estimation of the secondary path has no impact on the accuracy of the estimation.

2 Online Secondary Path Modeling Method

2.1 Existing Approach

Among the most recent online secondary path modelling methods [2,3,4,5,7,8], the method presented by Akhtar *et. al* [4] appears the best choice. Since the proposed method is based on Akhtar's algorithm, here this algorithm is described [4].

Consider Akhtar's method [4] shown in Fig. 1. The residual error signal *e(n)* of this algorithm is expressed as:

$$\begin{aligned} e(n) &= d(n) - y'(n) + v'(n) \\ y'(n) = s(n) * y(n) \ &, \ v'(n) = s(n) * v(n) \ , \end{aligned} \tag{1}$$

where *v(n)* is an internally generated white Gaussian noise, which is injected at the output of the control filter *W(z)*.

In this figure $\hat{S}(z)$ is the modelling FIR filter with length *M* that generates $\hat{v}'(n)$ expressed below:

$$\hat{v}'(n) = \hat{\mathbf{s}}^T(n)\mathbf{v}_M(n) . \tag{2}$$

As the figure shows, $\hat{v}'(n)$ generates the error signal for both the modelling filter $\hat{S}(z)$ and the control filter *W(z)* by subtracting from *e(n)*:

$$f(n) = [d(n) - y'(n) + v'(n)] - \hat{v}'(n) \ . \tag{3}$$

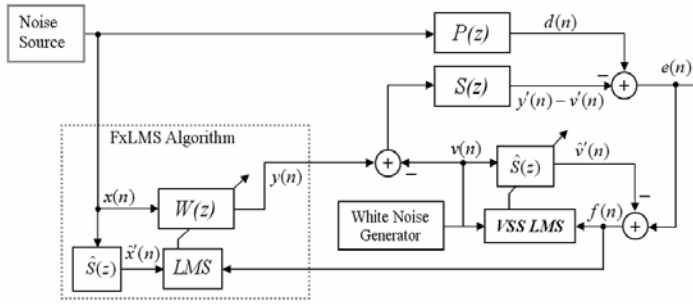


Fig. 1. Akhtar's method [4] for ANC system with online secondary path modelling

Coefficients of the modelling filter $\hat{S}(z)$ are updated as follows:

$$\hat{\mathbf{s}}(n+1) = \hat{\mathbf{s}}(n) + \mu_s(n) f(n) \mathbf{v}(n), \tag{4}$$

where $\mu_s(n)$ is the step-size parameter of the VSS-LMS algorithm which will be explained later.

Finally coefficients of the control filter *W(z)* are updated as below:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu_w(n) f(n) \hat{\mathbf{x}}'(n). \tag{5}$$

The input to the LMS algorithm is derived by filtering the reference signal through $\hat{S}(z)$:

$$\hat{x}'(n) = \hat{\mathbf{s}}^T(n) \mathbf{x}_M(n), \tag{6}$$

where $\mathbf{x}_M(n) = [x(n), x(n-1), ..., x(n-M+1)]^T$ is an M sample reference signal. The VSS-LMS algorithm is used to update modelling filter $\hat{S}(z)$ coefficients. For more detail on theory of this algorithm reader may refer to [4]. As we mentioned before, the modelling filter in (4) is updated using the step-size parameter ($\mu_s(n)$) of VSS-LMS algorithm and this parameter is calculated using the following three steps [4]:

- Initially, the power of error signals *e(n)* and *f(n)* are computed:

$$\begin{aligned} P_e(n) &= \lambda P_e(n-1) + (1-\lambda) e^2(n) \\ P_f(n) &= \lambda P_f(n-1) + (1-\lambda) f^2(n), \; 0.9 < \lambda < 1. \end{aligned} \tag{7}$$

- Then, the ratio of the estimated powers is obtained:

$$\begin{aligned} &\rho(n) = P_f(n) / P_e(n) \quad . \\ &\rho(0) \approx 1 \;,\; \lim_{n \to \infty} \rho(n) \to 0 \end{aligned} \tag{8}$$

- Finally, the step size is calculated as follows:

$$\mu_s(n) = \rho(n) \mu_{s_{\min}} + (1 - \rho(n)) \mu_{s_{\max}}, \tag{9}$$

where $\mu_{s_{\min}}$, $\mu_{s_{\max}}$ and λ are experimentally determined. Using VSS-LMS algorithm increases the modelling accuracy and correspondingly improves system performance. Indeed, Akhtar's method completely provided these features.

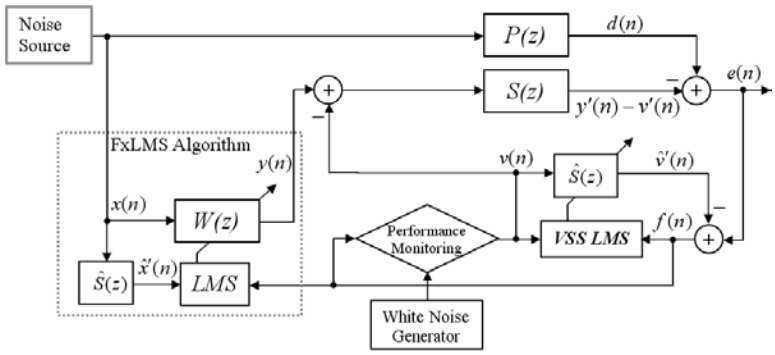


Fig. 2. Block diagram of the proposed feedforward ANC system

2.2 Proposed Method

As mentioned before, white noise with larger variance results in a better modelling accuracy and convergence rate [5]. However larger the variance increases the residual noise, which decreases performance of the system. Existing online secondary path modelling techniques [2,3,4,5,7,8] control the secondary path changes during system operation by continuous injection of white noise. Thus, these methods use white noise with a low variance to model the secondary path in order to maintain a lower residual noise in steady state. To use the advantage of large variance white noise and to adapt the system for secondary path changes, we propose a new system on the basis of Akhtar's method [4]. The proposed method modifies the Akhtar's algorithm to achieve a higher system performance.

Fig. 2 shows block diagram of the proposed system. To increase performance of the algorithm and to prevent the disadvantages of the white noise, we stop the VSS-LMS algorithm at the optimum point. This increases noise attenuation and allows benefiting the advantages of white noise with a large variance.

Here, the VSS-LMS algorithm is briefly described to show the way the optimum point is obtained. This algorithm is initially set to a small step size. During the process, μ_s is increased as the error signal *f(n)* decreases and vice versa. Consequently, once *W(z)* is slow in reducing *e(n)*, the step size remains small which results in a lower convergence rate. Hence, the modeling filter, $\hat{S}(z)$, converges to a good estimation when *f(n)* decreases. This happens when μ_s increases as high as $\mu_{s_{\max}}$. Thus, the injection of the white noise is stopped at the optimum point which is measured using:

$$\mu_{s_{\max}} - \mu_s < \alpha, \qquad 1\times10^{-5} < \alpha \le 1\times10^{-3}. \tag{10}$$

As shown in Fig. 2, this condition validity is monitored at the performance monitoring stage.

In some practical cases the secondary path may suddenly change. This event derives system to diverge. To prevent this effect, $\hat{S}(z)$ needs to be updated.

The proposed algorithm is designed in such a way that it can monitor the secondary path changes by the following expression:

$$20\log_{10}|f(n)| < 0 . \tag{11}$$

If the validity of the above equation does not satisfy, the system reactivates the VSS-LMS algorithm and injects white noise to remodel $\hat{S}(z)$. The same as before, the injection is stopped at the optimum point using (10).

The above procedure is repeated during the system operation to adapt the algorithm with characteristics of the environment.

Estimation of the secondary path can be obtained by using the off-line modeling method followed by an online modeling. However, as mentioned before, in some applications the primary noise exists even during the off-line modeling in which adversely affects the accuracy of the modeling filter. Therefore, with the advantages of the large variance white noise, there is no need of using off-line estimation of the secondary path in the proposed method as it is required in the existing methods [3, 4].

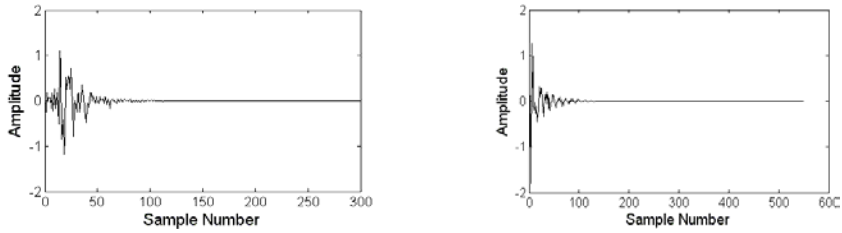
3 Simulations Results and Performance Evaluation

In this section the proposed ANC system is simulated using *Matlab* version *7.1*. In this simulation, we have used the primary path *P(z)* and secondary path *S(z)* of the experimental data provided in [6]. The impulse responses of the primary and secondary paths are shown in Fig. 3. Using these data, *P(z)* and *S(z)* are considered as FIR filters with tap–weight lengths 48 and 16 respectively. Rate of the sampling frequency in this simulation was 2 *KHz.*

Comprehensive experiments have been performed to find appropriate values for a fast and stable performance of the ANC system. Length of FIR filter $\hat{S}(z)$ for modelling the secondary path, and length of the adaptive filter *W(z)* used for the noise cancellation have been chosen 16 and 32, respectively.

In this simulations performance of the proposed method is compared with that of Akhtar's [4] and Eriksson's method [7]. We have performed simulations for two separate cases. In Case1, performance of the proposed method is evaluated in using both low and large variance white noises. Finally Case2 indicates effectiveness of the proposed algorithm in maintaining its performance against sudden changes of the secondary path behaviour. The parameters for the Akhtar's and proposed method are adjusted as $\mu_w = 5\times10^{-4}$, $\mu_{s_{\min}} = 75\times10^{-4}$, $\mu_{s_{\max}} = 25\times10^{-3}$ and $\lambda = 0.99$. For the proposed method we set α to 1.65×10^{-4}.

The parameters for Eriksson's method are adjusted using $\mu_w = 5\times10^{-4}$ and $\mu_s = 1\times10^{-2}$. To show the convergence rate and modelling accuracy of the system we use the relative modelling error given as:



(a) (b)

Fig. 3. Impulse response of the acoustic paths: (a) impulse response of the primary path *P(z)*, (b) impulse response of the secondary path *S(z)*

$$\Delta S(dB) = 10\log_{10}\left\{\frac{\sum_{i=0}^{M-1}[s_i(n)-\hat{s}_i(n)]^2}{\sum_{i=0}^{M-1}[s_i(n)]^2}\right\} \tag{12}$$

To signify performance of the system on noise reduction the following equation is used:

$$R = -10\log_{10}\left(\sum e^2(n) \Big/ \sum d^2(n)\right) \tag{13}$$

All the results shown in these cases are averaged on 10 experiments.

3.1 Case1

Here we evaluate the proposed method under the situations defined in [4] using white noise with two different variances. To set the initial value for $\hat{S}(z)$ ($\hat{s}(0)$), offline secondary path modeling is performed. The off-line modeling is stopped when the modeling error (12) has been reduced to -5 dB. In this experiment the reference noise is a narrowband signal comprising frequencies of *100*, *200*, *300*, and *400 Hz*. Its variance is adjusted to *2*, and a white noise with SNR of *30 dB* is added.

Fig . 4a shows that the performance curve of the proposed algorithm along with that of the reference curve raised to reach the optimum point, then by stopping the injection of white noise the former performance curve raises more rapidly compared with the reference algorithm. This indicates that the proposed method has a better performance in noise reduction compared with the existing approaches.

As Fig. 4b shows, the proposed method achieves a higher performance in using a larger variance white noise, in contradict, performance of the other methods is reduced by increasing the white noise variance. This is due to the non-stopping injection of white noise during the operation.

3.2 Case2

In this case, it is assumed that the secondary path transfer function is suddenly changed during the operation. Here we consider the reference signal used in Case1.

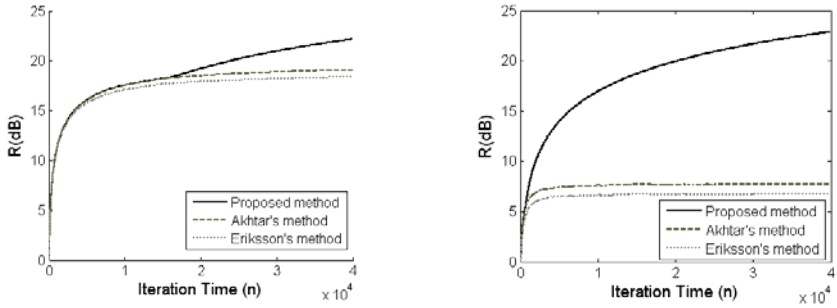


Fig. 4. Performance comparison between proposed method and other the existing methods. Noise reduction achieved by white noise with: (a) variance of 0.05, (b) variance of 0.8.

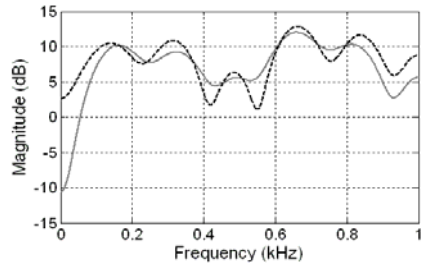


Fig. 5. Magnitude response of secondary path. (Solid line: Original path, Dashed line: Changed path at n=20,000).

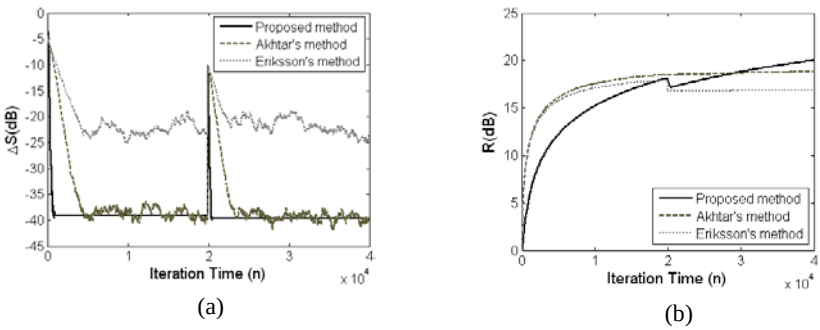


Fig. 6. Simulation results. (a) Relative modeling error versus iteration time *n*, (b) Noise reduction versus iteration time *n*.

Fig. 5 shows the magnitude response of the original and changed path. In this figure, the solid line represents secondary path at start point, $n = 0$, and the dashed line represents the changed path at iteration $n = 20000$. As can be resulted from Case1, a large variance white noise decreases performance of the other methods. Hence, we set two other methods at the conditioned described in [4]. In this case, the proposed method is evaluated using white noise with variance *0.8*, and setting initial value of

$\hat{S}(z)$ into zero. By using (12) and (13) the comparative results are shown in Fig. 6. As can be seen from these figures, without using offline estimation of secondary path, the proposed method approximately obtains the same results as Akhtar's method in modeling accuracy. The importance is that the proposed method achieves a higher noise reduction (R) and convergence rate compared to the other methods.

4 Conclusions

This paper proposed a new technique for online secondary path modelling in ANC systems to benefit from the injection of white noise. Simulation results for a single-channel feedforward ANC system demonstrated that the proposed method not only eliminates the disadvantages of the large variance white noise but also benefits from its advantages. The results indicate that the proposed method achieves a higher convergence rate, a more accurate modelling accuracy, and a better noise reduction performance compared with the existing approach.

References

1. Kuo, S.M., Morgan, D.R.: Active noise control: a tutorial review. Proc. IEEE 8(6), 943–973 (1999)
2. Kuo, S.M., Vijayan, D.: A secondary path modeling technique for active noise control systems. IEEE Trans. Speech Audio Proc. 5(4), 374–377 (1997)
3. Akhtar, M.T., Abe, M., Kawamata, M.: Modified-filtered-x LMS algorithm based active noise control system with improved online secondary path modeling. In: Proc. IEEE Intern. Mid. Symp. Circuits Systems, Hiroshima, Japan, pp. I-13–I-16, July 25–28 (2004)
4. Akhtar, M.T., Abe, M., Kawamata, M.: A Method for Online Secondary Path Modeling in Active Noise Control Systems. In: Proc. IEEE Intern. Symp. Circuits Systems (ISCAS 2005), pp. I-264–I-267, May 23–26 (2005)
5. Kuo, S.M., Vijayan, D.: Optimized Secondary Path Modeling Technique for Active Noise Control Systems. In: Proc. IEEE Asia-Pacific Conf. on Circuits and Systems, Taipei, Taiwan, pp. 370–375 (1994)
6. Kuo, S.M., Morgan, D.R.: Active Noise Control Systems-Algorithms and DSP Implementation. Wiley, New York (1996)
7. Eriksson, L.J., Allie, M.C.: Use of random noise for on-line transducer modeling in an adaptive active attenuation system. J. Acoust. Soc. Am. 85(2), 797–802 (1989)
8. Zhang, M., Lan, H., Ser, W.: Cross-updated active noise control system with online secondary path modeling. IEEE Trans. Speech Audio Proc. 9, 598–602 (2001)

Comparison of Linear and Nonlinear Models for Estimating Brain Deformation Using Finite Element Method

Hajar Hamidian[1], Hamid Soltanian-Zadeh[1,2], Alireza Akhondi-Asl[1], and Reza Faraji-Dana[3]

[1] Control and Intelligent Processing Center of Excellence (CIPCE), School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran
[2] Radiology Image Analysis Lab., Henry Ford Hospital, Detroit, MI 48202, USA
[3] School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran
h.hamidian@ece.ut.ac.ir, hszadeh@ut.ac.ir, a.akhundi@ece.ut.ac.ir, hamids@rad.hfh.edu, reza@ut.ac.ir

Abstract. This paper presents finite element computation for brain deformation during craniotomy. The results are used to illustrate the comparison between two mechanical models: linear solid-mechanic model, and non linear finite element model. To this end, we use a test sphere as a model of the brain, tetrahedral finite element mesh, two models that describe the material property of the brain tissue, and function optimization that optimizes the model's parameters by minimizing distance between the resulting deformation and the assumed deformation. Linear and nonlinear model assumes finite and large deformation of the brain after opening the skull respectively. By using the accuracy of the optimization process we conclude that the accuracy of nonlinear model is higher but its execution time is six time of the linear model.

Keywords: Finite element method, optimization process, brain model.

1 Introduction

Mechanical property of very soft tissue such as brain, liver, and kidney has been studied in recent years. This is because of applications such as surgical robot control system [1], surgical operation planning, and surgeon training systems based on the virtual reality techniques.

However, in a common neurosurgical procedure of the brain, it deforms after opening the skull, causing misalignment of the subject to the preoperative images such as magnetic resonance image (MRI) (Figure 1(a)-(b)) or computed tomography (CT) images [2], [3]. This phenomenon happens because of cerebrospinal fluid (CSF) leakage, dura opening, anaesthetics and osmotic agents, as well as conditions which are different from the normal state. Opening the scalp and CSF leakage cause the gravitational shift of the tissue due to disappearance of tension and pressure forces on the boundary condition of the brain [4], [5]. While the intraoperative imaging is the best

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 340–347, 2008.

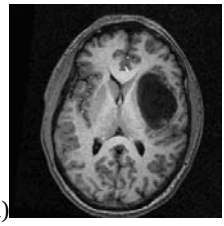
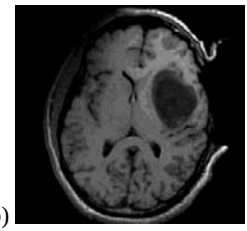
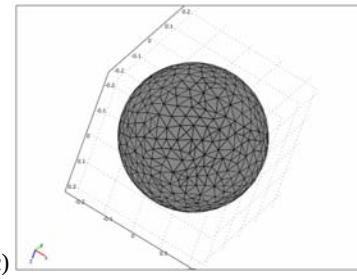
a)  b)  c) 

Fig. 1. MRI image (a) before opening the scalp, (b) after opening the scalp, (c) The result of meshing the volume

way to determine this deformation, intraoperative images suffer from the constraints of the operating room. Thus, spatial resolution and contrast of intra operative images are typically inferior to those of preoperative ones [6]. This problem can be solved by using biomedical models.

To this end, two models have been proposed in recent years, as described next. In 1999 K. Miller [7] suggested a model based on equation of equilibrium that related the covariant differentiation of stress with respect to the deformed configuration to body force per unit mass. In this model brain deformation is supposed to be large, brain tissue is treated as a hyper viscoelastic material and the stress–strain behavior of the brain tissue is non-linear [8], [9]. In 2002, M. Ferrant [10] proposed a mechanical model based on this principle that the sum of the virtual work from the internal strains is equal to the work from the external loads. In this formulation, brain deformation is supposed to be infinitesimal, brain tissue is treated as an elastic material, and the relation between strain and stress is linear [11].

In most practical cases, the above models utilize the finite element methods [12] to solve sets of partial differential equations governing the deformation behavior of the tissue. Using these methods, one can define the brain deformation assuming the brain's parameters are known. Previous works used approximate values of the brain parameters, which we also use in this work. We apply function optimization to optimize these parameters and minimize the distance between the resulting deformation and the supposed deformation.

In this paper, we use the above two models of the brain and optimize their parameters to match their resulting deformation with the assumed deformation. We then compare the two models using their resulting errors. In the next section, we explain the models and describe how to use meshing and boundary conditions for solving the problem using finite element methods and how to use function optimization to optimize their parameters. In Section 3, we explain the results of our implementation on a sphere as a model of the brain and compare the methods. Section 4 presents the conclusions of our work.

2 Materials and Methods

2.1 Construction of Finite Element Mesh

Within Finite Element Modeling (FEM) framework, the body on which one is working needs to be discretized using finite element mesh. By partitioning the object into

small elements, the equations are solved for each element, thereby solved for the whole object.

To this end, we use a sphere with a diameter of 22 Cm which is approximately the size of the brain. We also use FEMLAB3.3 to generate 4-noded tetrahedral mesh with Lagrange shape function (Figure 1(c)). This software generates the mesh automatically and also by changing its parameters, the user can change the mesh size.

2.2 Biomedical Models

As mentioned before, for determining the deformation of the brain, a model for the brain may be used. Such a model provides numerical formulations that can describe the behavior of the brain tissue. These formulations can be linear or non-linear. The linear model is simpler to implement [13], [14] and needs less time but a nonlinear model is more complicated and has better accuracy. In this section, we illustrate two models: one model describes the tissue behavior linearly and the other assumes the brain tissue to be nonlinear.

2.2.1 Linear Solid-Mechanic Model

In this model, the body is assumed to be a linear elastic continuum with no initial stresses or strains. The energy of the body's deformation caused by externally applied forces can be expressed as equation (1) [10].

$$E = \frac{1}{2}\int_{\Omega} \sigma^T \varepsilon \, d\Omega + \int_{\Omega} F^T u d\Omega, \tag{1}$$

ε is the strain vector that can be defined as equation (2) [10].

$$\varepsilon = \left(\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}, \frac{\partial u}{\partial z}, \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y}, \frac{\partial u}{\partial y} + \frac{\partial u}{\partial z}, \frac{\partial u}{\partial x} + \frac{\partial u}{\partial z}\right)^T = Lu \tag{2}$$

Also, σ is the stress vector and in the case of linear elasticity, with no initial stresses or strains, relates to the strain vector by the linear equation $\sigma = D\varepsilon$ [10]. The value of *D* depends on two material parameters: the Young modules and the Poisson ratios. The description of parameters can be found in [10]. Volumetric deformation of the brain is founded by solving equation (1) for the displacement vector *u*, which minimizes the energy function *E*. Numerical solution to this equation could be written in a global linear equation (3) [10].

$$Ku = -F \tag{3}$$

The solution of equation (3) provides us the deformation field that results from the forces applied to the body. We rely on the study of Ferrant et al in [10] and choose our initial coefficients (Young modules = 3 kPa, Poisson ratio = 0.45).

2.2.2 Non-linear Model

In this model the brain is supposed to be a single-phase continuum undergoing large deformations. In this analysis, the stresses and strains are measured with respect to the current configuration. Therefore, using Almansi strain and Cauchy stress, the virtual work principle can be written in the equation (4) [15].

$$\int_V \tau_{ij}\delta\varepsilon_{ij}dV = \int_V f_i^B \delta u_i dV + \int_S f_i^S \delta u_i dS, \tag{4}$$

The description of parameters can be found in [15]. As the brain undergoes finite deformation, current volume *V* and surface *S*, in the integration of equation (4) is unknown. Equation (4) forms a so-called weak formulation of the problem.

An alternative, so-called strong formulation that uses Einstein summation convention is given by differential equation (5) [16].

$$\nabla.\tau + \rho F = 0, \tag{5}$$

The description of parameters can be found in [16]. Equation (4), (5) must be supplemented by a formula that describes the mechanical property of the materials, relating the stress to the deformation of the body. There exists a variety of methods to solve integral equations (4) and strong formulation (5). Boundary Element Method uses the weak form but this method is not suitable for large deformations and nonlinear materials, rather for quasi-static small deformation. Therefore, in this paper we use the strong equation which is appropriate for our application.

As shown by [7], [17], the stress-strain behavior of the brain tissue is nonlinear. This model is suitable for low strain-rates typical for surgical procedures. In this paper we use the model suggested in equation (6) [7].

$$W = \int_0^t \left\{ \sum_{i+j=1}^{N} \left[C_{ij0} \left(1 - \sum_{k=1}^{n} g_k (1 - e^{-(t-\tau)/\tau_k}) \right) \right] \times \frac{d}{d\tau} [(J_1 - 3)^i (J_2 - 3)^j] \right\} d\tau \tag{6}$$

where τ_k is characteristic time, g_k is the relaxation coefficient, *N* is the order of polynomial in strain invariants, parameters C_{ij0} describe the instantaneous elasticity of the tissue, and J_1, J_2, and J_3 are strain invariants which are defined by equation (7) [7].

$$J_1 = Trace[B], J_2 = \frac{J_1^2 - Trace[B^2]}{2J_3}, J_3 = \sqrt{\det B} \tag{7}$$

Here, *B* is left Cauchy-Green strain tensor. We use the stationary form of the equation (6) because we solve the problem for the steady state form of deformation when the deformation of the brain is completed. The initial value of model's parameters are taken from [7] for *n*=2, *N*=2 as summarized in Table 1.

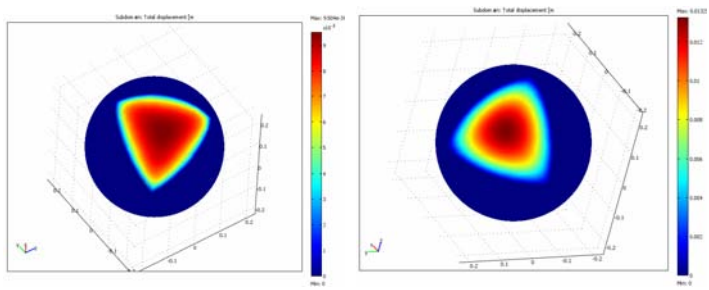
Table 1. List of material constants for model of brain tissue

Instantaneous Response	Characteristic time	Characteristic time
C100= 263 (Pa) C010= 263 (Pa) C110=0 (Pa) C020= 491 (Pa) C200 = 491 (Pa)	τ_1 = 0.5 (s)	τ_2 = 50 (s)
	instantaneous elasticity g_1=0.450	instantaneous elasticity g_2=0.365

2.3 Optimization Process

The brain parameters in a model are not certain for each individual and usually approximated parameters are used. In this paper, we use an optimization process to optimize these parameters to achieve the best results in comparison to the defined deformation. To this end, we choose a cost function that can be determined by the sum of displacements between the defined deformation of some specific points and the deformation of these points from the results of the two models. We use Maltlab optimization toolbox for the optimization procedure.

The displacement of special points can be defined by surgeon or imaging device such as MRI, CT, or spectroscopic camera. As mentioned before, the resolution of these images in operative room is not good and because of that it would be better to make images from special part, e.g., exposed surface of the brain or some parts that are more important for the surgeon like the tumor.



(a) (b)

Fig. 2. Result of the total displacement of subdomain, (a) for the first model, (b) for the second model

In both methods we have some parameters to optimize. In the first model, we can not determine the force applied to the exposed surface of the brain. This parameter can be defined by optimization process. Two parameters: Young modulus and Poisson's ratio are reported in previous papers but they are not necessarily the same for different patients so, these two parameters will be optimized.

In the second model, like the first one we can not determine the force applied to the exposed surface, so, this parameter would be determined by the optimization procedure. Also, the parameters in Table 1 except the characteristic time parameters are used as optimization parameters to minimize the cost function. This is because that we solve equation (6) in steady state, so, these parameters are not used. Also, in this state the sum of g_1 and g_2 are used. By using more defined points, the accuracy of our model can be improved and the estimated displacement for points inside the brain such as tumor would become more precise.

2.4 Boundary Conditions

For solving partial differential equations, we need some boundary condition. As mentioned before, for testing our method, we use a sphere as a simple model for the brain and for modeling the craniotomy, we assume one section on the sphere to be exposed. So, this section is free and the rest of them are fixed.

For the first model, we have conditions for displacement variable and F (force per unit). The boundary nodes that are not exposed are fixed. We use $F=u$ for boundary conditions of fixed boundary nodes because the elements of the rigidity matrix K in equation (3) that the deformation is supposed to be known need to be set to zero, and the diagonal elements of these rows to one. More detailed can be found in [10].

Also, for the second model, the boundary condition for F in addition to displacement (u) must be determined. Because this model is nonlinear, we can not summarize the equations to one equation like equation (3). So, we can not determine it and it will be an optimization parameter.

By using these conditions for implementing the models, the deformation of the whole brain can be determined and sued to determine some important part like tumor.

3 Results

For modeling the brain, we use a sphere with the diameter of 22 Cm that is approximately the size of the brain. To show the skull opening, we assume that one section of this sphere is exposed. We assume a model with special parameters. Then we change the parameters and use the optimization process to estimate the assumed model's parameters by using displacement of some points in the assumed model. This can show us if the brain can be modeled with a specific model and if our proposed method can accurately estimate that model. For each model, we do this process and compare the accuracy of models using error of the optimization function and the error of displacement of some other points that we do not use in the optimization process. To implement the models, we use the FEMLAB3.3 software which is based on the finite element methods for solving partial differential equations. This software has visualization, meshing, problem solving, and strong post processing modules. Figure 2 show the result of the two models. As can be seen, both models show the brain deformation continuously. The first model optimization mean error is 0.1172 mm, and the mean error for other points that we do not use in optimization process is 0.2731 mm. The second model optimization mean error is 0.0683 mm, and the mean error for other points that we do not use in optimization process is 0.1915 mm. Therefore, the accuracy of the second model is higher than the first model. Tables 2-3 show the assumed parameters and the estimated parameters for both models. As can be seen, the estimation parameters for the nonlinear model are closer to the assumed parameters than the linear model. Thus, the optimization process can estimate the parameters of nonlinear model more accurately than linear model.

Table 2. Assumed and estimated parameters for the linear model

	Young modulus	Poisson's ratio	Force
Assumed	0.45	3000	Fx=-1500 Fy=1500 Fz=1500
Estimated	0.45 ± 0.0056	3000 ± 175.6	Fx=-1500 ± 90.8 Fy=1500 ± 87.9 Fz=1500 ± 93.2

Table 3. Assumed and estimated parameters for the nonlinear model

	C100	C010	C200	C020	C110	g_1+g_2	Force
Assumed	263	263	491	491	0	0.815	Fx=-300 Fy= 300 Fz= 300
Estimated	263 ±4.0401	263 ±7.8404	491 ±14.4139	491 ±13.4578	0 ±0.001	0.815 ±0.0083	Fx=-300 ± 0.173 Fy= 300 ± 0.028 Fz= 300 ± 0.800

However, it must be considered that the execution time of the nonlinear model is approximately six time of the linear model. This happen because the first model is linear but the second model is nonlinear and more complicated than the first one. For implementation process we use computer with 3.6 GHz dual core CPU and 4GB RAM. In summary, the second model estimates the parameters more accurately but it takes much more time run due to its complication.

It must be mentioned that this paper is the first work of authors in this field.

4 Conclusion

Model selection is an important step in obtaining valid results. To this end, we choose two linear and nonlinear mechanical models that describe the mechanical property of the brain based on finite and large deformation respectively and implement them on a sphere. The linear model is based on this principle that the sum of the virtual work from the internal strains is equal to the work from the external loads and the nonlinear model is based on the equation of equilibrium that relates the covariant differentiation of stress to the body force. The nonlinear model is more accurate than the linear model and optimization process can estimate its parameters more accurately. This is because the nonlinear model has more parameters and also nonlinear model is more flexible than linear model. However, the nonlinear model is too complicated and its implementation time is six time of the linear model. Therefore, depending on the condition of surgery, both models have sufficient accuracy and can be used. By using the result of implementing these models on a sphere, we can select the best model for estimating the deformation of the brain.

References

1. Wittek, A., Miller, K., Laporte, J., Kikinis, R., Warfield, S.: Computing reaction forces on surgical tools for robotic neurosurgery and surgical simulation. In: CD Proceedings of Australasian Conference on Robotics and Automation ACRA, Canberra, Australia (2004)
2. Miller, K.: Method of testing very soft biological tissues in compression. J. Biomechanics 38, 153–158 (2000)

3. Miga, M.I., Paulsen, K.D., Hoopes, P.J., Kennedy, F.E., Hartov, J.A., Roberts, D.W.: In Vivo quantification of a homogeneous brain deformation model for updating preoperative images during surgery. IEEE Trans. Biomed. Eng. 47(2), 266–273 (2000)
4. Dumpuri, P., Thompson, R.C., Dawant, B.M., Cao, A., Miga, M.I.: An atlas-based method to compensate for brain shift: Preliminary results. Medical Image Analysis 11, 128–145 (2007)
5. Platenik, L.A., Miga, M.I., Roberts, D.W., Lunn, K.E., Kennedy, F.E., Hartov, A., Paulsen, K.D.: In vivo quantification of retraction deformation modeling for updated image-guidance during neurosurgery. IEEE Trans. Biomed. Eng. 49(8), 823–835 (2002)
6. Clatz, O., Delingette, H., Talos, I.F., Golby, A.J., Kikinis, R., Jolesz, F.A., Ayache, N., Warfield, S.K.: Robust nonrigid registration to capture brain shift from intraoperative MRI. IEEE Trans. Med. Imag. 24(11), 1417–1427 (2005)
7. Miller, K.: Constitutive model of brain tissue suitable for finite element analysis of surgical procedures. J. Biomech. 32, 531–537 (1999)
8. Miller, K.: How to test very soft biological tissues in extension? J. Biomech. 34, 651–657 (2001)
9. Miller, K., Chinzei, K.: Mechanical properties of brain tissue in tension. J. Biomech. 35, 483–490 (2002)
10. Ferrant, M., Nabavi, A., Macq, B., Black, P.M., Jolesz, F.A., Kikinis, R., Warfield, S.K.: Serial registration of intraoperative MR images of the brain. J. Med. Imag. Analysis 6, 337–359 (2002)
11. Ferrant, M., Nabavi, A., Macq, B., Jolesz, F.A., Kikinis, R., Warfield, S.K.: Registration of 3-D intraoperative MR images of the brain using a finite element biomechanical model. IEEE Trans. Med. Imag. 20, 1384–1397 (2001)
12. Bathe, K.J.: Finite element procedures. Prentice-Hall, Englewood Cliffs (1996)
13. Miga, M.I., Paulsen, K.D., Lemery, J.M., Eisner, S.D., Hartov, A., Kennedy, F.E., Roberts, D.W.: Model-Updated Image Guidance: Initial clinical experiences with gravity-induced brain deformation. IEEE Tran. Med. Imag. 18(10), 866–874 (1999)
14. Lunn, K.E., Paulsen, K.D., Liu, F., Kennedy, F.E., Hartov, A., Roberts, D.W.: Data-guided brain deformation modeling: evaluation of a 3-D adjoint inversion method in porcine studies. IEEE Trans. Biomed. Eng. 53(10), 1893–1900 (2006)
15. Wittek, A., Miller, K., Kikinis, R., Warfield, S.K.: Patient-specific model of brain deformation: Application to medical image registration. J. Biomech (in press, 2006)
16. Wittek, A., Kikinis, R., Warfield, S.K., Miller, K.: Computation using a fully nonlinear biomechanical model. In: Duncan, J.S., Gerig, G. (eds.) MICCAI 2005. LNCS, vol. 3750, pp. 583–590. Springer, Heidelberg (2005)
17. Miller, K., Chinzei, K., Orssengo, G., Bednarz, P.: Mechanical properties of brain tissue in-vivo: experiment and computer simulation. J. Biomech. 33, 1369–1376 (2000)

Context-Dependent Segmentation of Retinal Blood Vessels Using Hidden Markov Models

Amir Pourmorteza[1], Seyed Hamid Reza Tofighi[1], Alireza Roodaki[1], Ashkan Yazdani[1], and Hamid Soltanian-Zadeh[1,2,3]

[1] Control and Intelligent Processing Center of Excellence, Department of Electrical and Computer Engineering, University of Tehran, Tehran, Iran
a.pourmorteza@ece.ut.ac.ir, hszadeh@ut.ac.ir
[2] School of Cognitive Sciences, Institute for Studies in Theoretical Physics and Mathematics (IPM), Tehran, Iran
[3] Image Analysis Lab., Department of Radiology, Henry Ford Hospital, Detroit, Michigan, USA
hamids@rad.hfh.edu

Abstract. Hidden Markov Models (HMMs) have proven valuable in segmentation of brain MR images. Here, a combination of HMMs-based segmentation and morphological and spatial image processing techniques is proposed for the segmentation of retinal blood vessels in optic fundus images. First the image is smoothed and the result is subtracted from the green channel image to reduce the background variations. After a simple gray-level stretching, aimed to enhance the contrast of the image, the feature vectors are extracted. The feature vector of a pixel is formed from the gray-level intensity of that pixel and those of its neighbors in a predefined neighborhood. The ability of the HHMs to build knowledge about the transitions of the elements of the feature vectors is exploited here for the classification of the vectors. The performance of the algorithm is tested on the DRIVE database and is comparable with those of the previous works.

Keywords: Hidden Markov models (HMM), context-dependent image segmentation, retinal images.

1 Introduction

Early symptoms of systematic diseases can be detected by the assessment of retinal images. Eye is considered as a window to the retinal vasculature, where the influence of the factors that affect the human body vasculature can be observed *in vivo* non-invasively. Furthermore, inspection of optic fundus photographs [1]-[4], and flourocein images [2] may help to diagnose and monitor the progress of general diseases such as diabetes, hypertension, arteriosclerosis, cardiovascular diseases, stroke and eye diseases like retinopathy of prematurity [1]-[3]. Thus, the measurement and analysis of retinal blood vessels is of diagnostic value for a wide range of pathological states. However, manual analysis of the complex retinal blood vessel trees in fundus images is a tiresome and laborious task, and as the number of images increases, it may even be impossible.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 348–355, 2008.

Segmentation of the vessels from the background is an initial requirement for the automatic assessment of retinal images. Many automatic algorithms are proposed for the segmentation of retinal blood vessels. According to [9], methods for detecting blood vessels generally fall into three categories: kernel-based, tracking based, and classifier-based.

In this paper, we will introduce a novel method for the automated segmentation of blood vessels. We start with simple kernel-based preprocessing operations followed by a contrast enhancement step. Then a Hidden Markov Model-based classification is used to assign the pixels to vessel or background classes.

Recently, Hidden Markov Models (HMMs) are used for MRI brain segmentation in [5]. Markov Random Fields (MRF), Hidden Markov Random Fields (HMRF), and Hidden Markov Models, take advantage of the relative information of the vectors and fall under the category of context-dependent classifiers. Kernel-based and thresholding methods, simply assume no relation between different classes, i.e. once a feature vector (corresponding to a pixel) is assigned to a class, the next vector may be assigned to any other class. Whereas in context-dependent classification, the class to which a vector is assigned, depends (a) on its own value, (b) on the value of other feature vectors and (c) on the existing relation among various classes.

MRFs and HMRFs were introduced in several segmentation frameworks [11]-[14]. While these frameworks encode the dependency between the pixel to be segmented and its first-degree neighbors, they are computationally intensive and therefore, they are not welcome in medical environments [5]. In contrast, HMMs have proven valuable in Automatic Speech Recognition (ASR) and MRI brain segmentation tasks [5],[6].

The primary goal of this paper is to report a method which combines kernel-based techniques with state-of-the-art HMM-based segmentation. The algorithm is evaluated using the images obtained from the publicly available DRIVE database [16]. This combined method allows for results comparable with manually segmented images as well as with those reported by other authors. A brief introduction to HMMs and its training algorithms is provided in section 2. Section 3 details the preprocessing and feature extraction algorithm. Section 4 includes the segmentation algorithm and necessary post-processing step to improve the results. Finally, experimental results of the proposed method are presented in section 5.

2 Hidden Markov Models

Similar to the work done in [5], the fundamental idea of this paper relies on the ability of the underlying HMM to encode the knowledge about the input data vectors or sequences that reflect the characteristics of the image to be segmented e.g. intensity information about the pixel and its neighborhood.

HMMs are made of different states statically bound by transition probabilities. An HMM is characterized by a set of internal states, the transition probabilities among the states in response to an input symbol from the sequence, and the emission probabilities of symbols from the different states[5]. The knowledge is built in the form of the transition and the emission probabilities of the states that are trained during the learning stage. HMMs assume that the states are hidden and cannot be observed at the output stage. Instead, only the output emitted from the states are observable, without

knowing which states emitted those outputs. From this point of view, the HMM is regarded as a symbol generating process in which the observations are viewed from the outside without knowing which state emitted them. Here, similar to what has been previously done in [5] HMMs are viewed from a different perspective. In using HMMs for blood vessel segmentation, the goal is to find the best state sequence that might have produced a specific output.

Hence, during the training, the goal of the training algorithm is to increase the output probability of the input sequences representing a class of tissue [5]**.**

The transition and emission probabilities are updated in a manner that maximizes the output probability of a given class. As a result, the relationship between the elements of a sequence (voxels of a neighborhood) is encoded in the transition and emission matrices.

The concept of Minimum Classification Error is used to update the parameters of both discrete and continuous mixture of Gaussian distributions in [5]. In [6], an Expectation Maximization (EM) algorithm is utilized to optimize the parameters of a HMM which has Gaussian pdfs for emission probabilities. Here, the algorithm used in [6] is selected. In our model, the sequence of observations is nothing more than a sequence of pixel intensities. There are many ways to generate this observation sequence. One reasonable way is to consider a 3×3 neighborhood of a certain pixel and put the intensity of the pixel along with the intensities of its neighbors in a 9×1 feature vector. Other choices of neighborhoods are also possible. The increase in size of the neighborhood may adversely affect the accuracy due to smoothing effect of large neighborhoods [5].

3 Preprocessing

The goal of preprocessing phase is to reduce the unwanted effects of noise and background intensity variations, and enhance the contrast between the vessel and non-vessel pixels prior to segmentation. Since the HMMs will have to learn respond to feature vectors from different images, the images must share nearly similar brightness and contrast characteristics. By compensating for these through and across data set variations, one can improve the performance of the segmentation process.

First step is to choose a monochromatic image that shows high contrast between vessel and non-vessel pixels. A Monte Carlo simulation of retinal vessel profiles [15] predicts that light that is observed from a retinal vessel at green band on a RGB color image is predominantly backscattered from the vessel or transmitted once through the vessel [2]. This is confirmed by experimental results [1],[4],[8]. Hence we have selected the green channel representation. In an attempt to compensate for the contrast and brightness variations among the images of the data set, the mean and standard deviation of the histograms of the images are changed so that they nearly match predefined values of mean=0.5 and S=0.075.

To compensate for the background intensity variations, a smoothing kernel is convolved with the green channel. This convolution gives an approximation of the background variations. Next, this approximation is subtracted from the green image. The result is a background normalized image as can be seen in fig.1(c)**.** Size of the smoothing kernel is empirically selected to be 11×11.

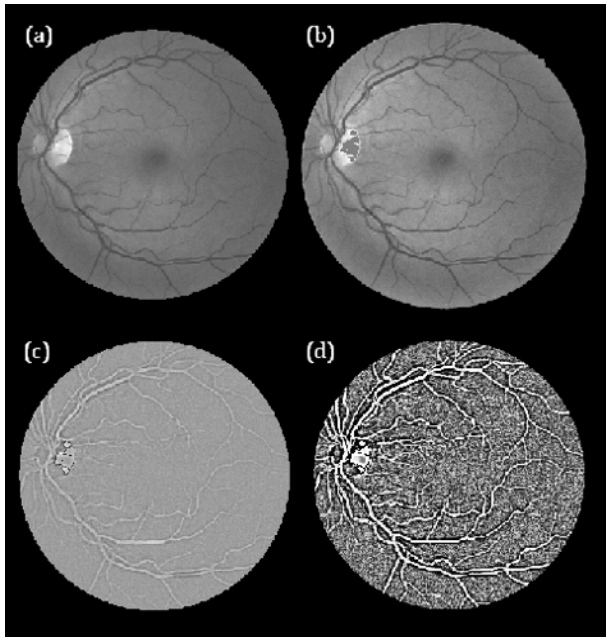


Fig. 1. (a) green channel of original image, (b) image with corrected mean and variance values, (c) background normalized image, and (d) contrast enhanced image

Histogram of the background normalized image resembles a Gaussian distribution with a small variance in which most of the pixels have midrange gray-levels. In an attempt to enhance the global contrast, the histogram is stretched by a sigmoid function. Results of this step are illustrated in fig. 1.

4 Segmentation and Post-processing

The notion behind HMMs-based segmentation [5] is that, by training a HMM with feature vectors representing a special class of data, the model learns to produce higher outputs when a vector of that class is presented to the model as an input sequence. Hence, the segmentation algorithm is straightforward. A HMM is trained for each class of data, the output probability of every HMM for the feature vector representing the pixel to be classified is then computed using the trained transition matrices and emission vectors that encode the relative dependency of the elements of the feature vector. The segmentation is then done, by simply assigning the pixel to the class associated with the HMM showing the highest output probability.

In this paper, two HMMs are trained. The first model is trained using feature vectors randomly chosen from vessel pixels of manually segmented images. The second HMM is trained using non-vessel feature vectors. Feature vectors corresponding to every non-black pixel of the image is then presented to the HMMs. The output probabilities of the HMMs are represented in two likelihood maps (see figure 2). The maps are then compared pixel by pixel and the pixels whose vessel likelihoods are

greater are segmented as vessel. The HMMs do not learn every possible form of blood vessel or background sequences. This is because the training features are chosen randomly.

Therefore, some sequences may never show up in the training step. This fact and the presence of noise and contrast variations, result in falsely detected blood vessels. To eliminate these, a post-processing step is necessary. Morphological operations such as erosion and dilation can be very helpful in this step. First, the image is eroded using a structural element of a certain size. Next, connected components smaller than a specified size are removed. Finally the image is dilated using the same structural element that was used for eroding (see fig. 3.).

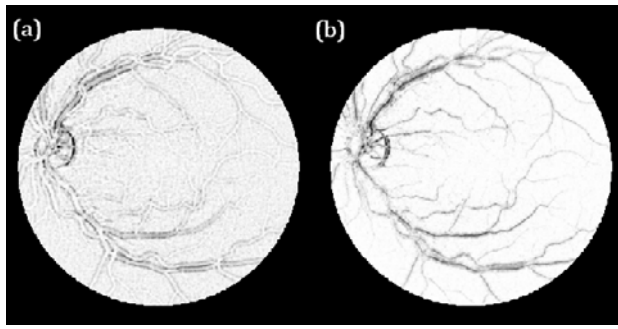


Fig. 2. Likelihood maps of (a) vessel and (b) non-vessel classes

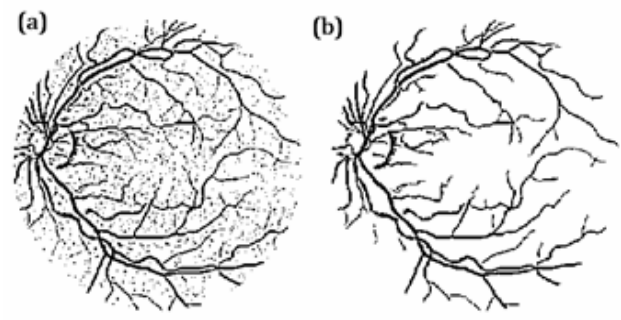


Fig. 3. Result of segmentation step before (a) and after (b) morphological post-processing

5 Experimental Results

Our proposed method was tested on images of the publicly available DRIVE database [16]. The DRIVE database contains 40 color images of the retina, with 565 × 584 pixels and 8 bits per color channel, in LZW compressed TIFF format. The images were originally captured from a Canon CR5 nonmydriatic 3 charge-coupled device (CCD) camera and were initially saved in JPEG-format. The database also includes binary images with the results of manual segmentation. These binary images have already been used as ground truth for performance evaluation of several vessel

segmentation methods [16]. The 40 images were divided into a training set and a test set by the authors of the database. The results of the manual segmentation are available for all the images of the two sets. For the images of the test set, a second independent manual segmentation also exists.

Segmentation accuracy is selected as performance measure to compare our results with previous retinal vessel segmentation algorithms. The accuracy is estimated by the ratio of the total number of correctly classified points (sum of true positives and true negatives) by the number of non-black points in the image. The ground proof for computing the performance measures was the manual segmentation result. The values for the fraction of pixels erroneously classified as vessel pixels, false positive ratio (FPR), and the percentage of pixels correctly classified as vessel pixels, true positive fraction (TPR), are also reported. Two experiments were carried out to evaluate the performance of the proposed algorithm.

Table 1. Performance of vessel segmentation methods

Method	Accuracy	TPR	FPR
2nd human observer	0.9473	0.7761	0.0275
Mendonça et al. [1]	0.9452	0.7344	0.0236
Jiang et al. [4]	0. 8911	not reported	
Staal et al. [1]	0.9442	0.7194	0.0227
Niemeijer [1]	0.9417	0.6898	0.0304
Soares et al. [4]	0.9466	not reported	
First Experiment	0.9388	0.7342	0.0332
Second Experiment	0.9401	0.7492	0.0384

In our first experiment, a simple global thresholding algorithm was applied on the test set of the DRIVE database. The images were initially pre-processed according to the algorithm discussed in section 3. Next a global thresholding algorithm was applied on the data set. The threshold level was empirically chosen to be 0.6. Then, we used the first two images of the training set to train two HMMs with 2000 feature vectors randomly selected for each class of pixels using the labels of the results of the manual segmentation. Each HMM had 9 hidden states and the emission probabilities were approximated by mixtures of 15 Gaussian distributions. Next, the images of the test set were segmented followed by erosion, small connected component removal, and dilation to remove the falsely detected vessel pixels. To further improve the results, the union of the globally thresholded image and the result of the HMM-based segmentation were chosen as the final result, whenever the accuracy of the segmentation step was not acceptable.

The second experiment aimed to decrease the computation time of the segmentation step. The whole algorithm was the same as for the first experiment, except that the output likelihood maps of the HMMs were formed in a 4-connected manner, i.e. starting from the first pixel of the image, its 4-connected neighbors had the same likelihood as the pixel. As a result, the computational time was reduced to 20% of the first experiment. Consequently, once a pixel is chosen to be in class 1, its neighbors

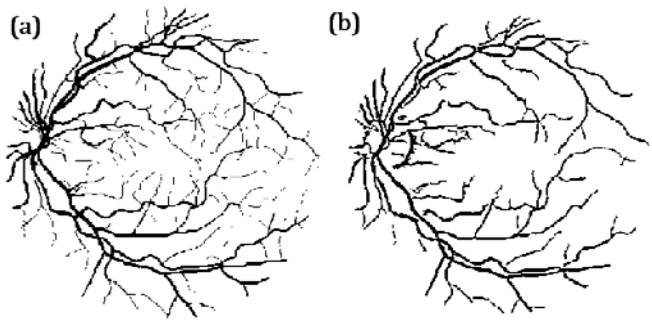


Fig. 4. Results from - (a) human observer, (b) our algorithm

are also chosen, resulting in a 4 connected segmented image which in turn, leads to better TPR since the number of small gaps between the connected components of the vessel class is reduced.

6 Discussion and Conclusion

Our method proved the ability of Hidden Markov Models in retinal blood vessel segmentation. The results are particularly comparable to previous works in which classifier-based methods are used. In [4], all 20 images of the training set of the DRIVE database were used for training, with about one million feature vectors. Whereas in our work, only 2000 feature vectors, extracted from 2 images, were used. Thus, the training time was reduced to approximately 1 hour, which is 9 times less than the time reported in [4]. However, the accuracies of these two methods are approximately equal. The simplicity of the feature vectors is also remarkable. Unlike [4] in which feature vectors are formed by Gabor transforms, our feature vectors are merely made of intensities of the pixels in a predefined neighborhood.

The downside to our algorithm is that the information used for training the HMMs is not accurate. In fact, the TPR and FPR for the human observers are 0.7761 and 0.0275, respectively. Therefore, some training vectors are wrongly presented to the HMMs and the models learn the wrong combination of pixel intensities. Another drawback of the proposed algorithm is that the EM algorithm used for the training of the models, may converge to local minima, resulting in poorly trained HMMs. In [5] the authors claim that under certain conditions, the MCE algorithm may achieve the global minimum with higher probability. The choice of number of hidden states and Gaussian distributions is also open to further investigation.

Acknowledgment

The authors would like to thank the authors of DRIVE database for making their data publicly available. We would also like to thank Kevin Murphy for making his HMM-toolbox for MATLAB available.

References

1. Mendonça, M., Campilho, A.: Segmentation of Retinal Blood Vessels by Combining the Detection of Centerlines and Morphological Reconstruction. IEEE Trans. Med. Imag. 147, 1200–1213 (2006)
2. Martinez-Prez, M.E., Hughes, A.D., Thom, S.A., Bharath, A.A., Parker, K.H.: Segmentation of Blood Vessels from Red-Free and Fluoroscein Retinal Images. Med. Image. Anal. 11, 47–61 (2007)
3. Vermeer, K.A., Vos, F.M., Lemij, H.G., Vossepoel, A.M.: A Model Based Method for Retinal Blood Vessel Detection. Comput. Biol. Med. 34, 209–219 (2004)
4. Soares, J.V.B., Leandro, J.J.G., Cesar Jr., R.M., Jelinek, H.F., Cree, M.J.: Retinal Vessel Segmentation Using the 2-D Gabor Wavelet and Supervised Classification. IEEE Trans. Med. Imag. 25, 1214–1222 (2006)
5. Ibrahim, M., John, N., Kabuka, M., Younis, A.: Hidden Markov Models-Based 3D MRI Brain Segmentation. Image. Vision. Comput. 24, 1065–1079 (2006)
6. Solomon, J., Butamen, J.A., Sood, A.: Segmentation of Brain Tumors in 4D MR Images Using the Hidden Markov Model. Comput. Meth. Prog. Bio. 84, 76–85 (2006)
7. Resch, B.: Hidden Markov Models. A Tutorial for the Course Computational Intelligence, `http://www.igi.tugraz.at/lehre/CI`
8. Feng, P., Pan, Y., Wei, B., Jin, W., Mi, D.: Enhancing Retinal Images by the Contourlet Transform. Pattern Recogn. Lett. 28, 516–522 (2007)
9. Hoover, A., Kouznetsova, V., Goldbaum, M.: Locating Blood Vessels in Retinal Images by Piecewise Threshold Probing of a Matched Filter Response. IEEE Trans. Medical Imaging 19, 203–219 (2000)
10. Chaudhuri, S., Chatterjee, S., Katz, N., Nelson, M., Goldbaum, M.: Detection of Blood Vessels in Retinal Images Using Two-Dimensional Matched Filters. IEEE Trans. Med. Imag. 8, 263–269 (1989)

Retinal Vessel Extraction Using Gabor Filters and Support Vector Machines

Alireza Osareh and Bita Shadgar

Shahid Chamran University, Ahvaz, Iran
{Alireza.Osareh,Bita.Shadgar}@scu.ac.ir

Abstract. Blood vessel segmentation is the basic foundation while developing retinal screening systems, since vessels serve as one of the main retinal landmark features. This paper proposes an automated method for identification of blood vessels in color images of the retina. For every image pixel, a feature vector is computed that utilize properties of scale and orientation selective Gabor filters. The extracted features are then classified using generative Gaussian mixture model and discriminative support vector machines classifiers. Experimental results demonstrate that the area under the receiver operating characteristic (ROC) curve reached a value equal to 0.974. Moreover, it achieves 96.50% sensitivity and 97.10% specificity in terms of blood vessels identification.

Keywords: Retinal Blood Vessels, Gabor Filters, Image Segmentation, Support Vector Machines.

1 Introduction

Diabetes is a disease that affects about 5.5% of the population worldwide, a number that can be expected to increase significantly in the coming years [1]. About 10.0% of all diabetic patients have retinopathy, which is the primary cause of blindness in the working population. Since this type of blindness can be prevented with proper treatment at its early stage, the World Health Organization advises yearly screening of patients. Thus, an automatic system can facilitate this screening process.

The blood vessels network is an important anatomical structure in the human retina. Several vascular diseases, such as diabetic retinopathy, have manifestations that require analysis of the vessels network. In other cases, e.g. pathologies like retinal microaneurysms and hemorrhages, the performance of automatic detection methods may be improved if regions containing vessels can be excluded from the analysis [2]. Indeed, the position, size and shape of the vessels provide information which can be used to locate the optic disk and the fovea (central vision area).

So far several methods have been developed for vessel segmentation, but visual inspection and evaluation by ROC analysis shows that there is still room for improvement [3]. Previous works on blood vessel detection can be mainly divided into 3 categories: window-based [4], classifier-based [5] and tracking-based [6]. Window-based approaches, such as edge detection, estimate a match at each pixel for a given model against the pixel's surrounding window. In [4], the cross section of a retinal vessel was modeled by a Gaussian shaped curve, and then detected using rotated matched filters.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 356–363, 2008.

Classifier-based algorithms proceed in two steps. First, a low-level algorithm produces a segmentation of spatially connected regions. These candidate regions are then classified as being vessel or not vessel. In [5], regions segmented by the method [4] were classified as vessel or not based on many properties, such as their response to a classic operator.

Tracking-based approaches utilize a profile model to incrementally step along and segment a vessel. In [6], the tracking method was driven by a fuzzy model of a one-dimensional vessel profile. One drawback to these approaches is their dependence upon unsophisticated methods for locating the starting points, which must always be either at the optic nerve or at subsequently detected branch points.

In this paper, we propose a novel vessel segmentation approach to efficiently locate and extract blood vessels in color retinal images. Our scheme consists of three major steps: multiscale analysis using Gabor filters, feature extraction and classifying the image pixels by Gaussian mixture models (*GMMs*) and support vector machines (*SVMs*) classifiers. Finally, the accuracy of our optimum classifiers are evaluated using ROC curves analysis and the sensitivity and specificity measurements.

In section 2, we will describe the properties of our images, and present a major in-depth review of the algorithm including application of Gabor filters and classification schemes. Indeed, the experimental evaluation and results are presented and discussed in this section, followed by our conclusions in section 3.

2 Materials and Methods

2.1 Image Acquisition

Many of the previous published vessels detection methods have not been evaluated on large datasets or fail to give satisfactory results for large numbers of images as encountered in a routine clinical screening program. Here, we implement and test our proposed scheme against the DRIVE dataset images [7]. This dataset contains 40 digital retinal images where each image was captured using 24 bit per pixel (standard RGB) at 565 x 584 pixels.

Retinal images can be either normal or abnormal. See Figure 1 for an example of both a normal and an abnormal retinal image. As can be seen, normal image consists of vessels, optic disc, fovea and the background, but the abnormal image has multiple artifacts of distinct shapes and colors (microaneurysms, hemorrhages and exudates) caused by different diseases such as diabetic retinopathy [8]. The image dataset is carefully labeled by an expert, to produce ground truth manual vessels segmentation.

The retinal image pixels are considered as objects represented by their feature vectors, so that we can apply statistical classifiers in order to classify the pixels. Here, we assume a binary multi-dimensional classification approach to distinguish the vessel pixels from other anatomical-pathological structures and artifacts, which we refer to collectively as non-vessels. Our chosen data for the learning stage consists of typical pixels, which are representative of our classification problem.

(a) (b)

Fig. 1. Typical normal and abnormal retinal images. (a) a normal image that shows blood vessels, optic disc and fovea components, (b) an abnormal image with exudate and hemorrhage pathologies.

2.2 Two-Dimensional Gabor Filters

Gabor filters are powerful tools that have been widely used for multi-scale/multi-directional analysis in image processing. These filters have specifically shown high performance as feature extractors for discrimination purposes [9]. Due to directional selectiveness capability in detecting oriented features and fine tuning to specific frequencies and scale, these filters act as low-level oriented edge discriminators. Mathematically, a two-dimensional (2D) Gabor function, *g*, is the product of a 2D Gaussian and an exponential function which can be given by:

$$g_{\theta,\lambda,\sigma_1,\sigma_2}(x,y) = \exp\left\{-\frac{1}{2}M(x,y)^T\right\}\exp\left\{\frac{j\pi}{\lambda}(x\cos\theta + y\sin\theta)\right\} \tag{1}$$

where $M = diag\left(\sigma_1^{-2}, \sigma_2^{-2}\right)$. The parameter θ represents filter orientation, λ is the filter wavelength which modify the sensitivity to high/low frequencies, and σ_1 and σ_2 characterize the filter standard derivations which represent scale value at orthogonal directions. When the Gaussian part is symmetric, we obtain the following isotropic Gabor function:

$$g_{\theta,\lambda,\sigma}(x,y) = \exp\left\{-\frac{x^2+y^2}{2\sigma^2}\right\}\exp\left\{\frac{j\pi}{\lambda}(x\cos\theta + y\sin\theta)\right\} \tag{2}$$

However, with this parameterization the Gabor function does not scale uniformly, when σ changes. Thus, it is preferable to use a new parameter $\gamma = \lambda/\sigma$ instead of λ so that a change in σ corresponds to a true scale change in the Gabor function. Also, it is convenient to apply a 90° counterclockwise rotation to Equation (2), such that θ expresses the orthogonal direction to the Gabor function edges. Therefore, in the reminder of the paper we use the following definition for the Gabor functions:

$$g_{\theta,\gamma,\sigma}(x,y) = \exp\left\{-\frac{x^2+y^2}{2\sigma^2}\right\}\exp\left\{\frac{j\pi}{\gamma\sigma}(x\sin\theta - y\cos\theta)\right\} \tag{3}$$

By convolving a Gabor function with image patterns *f(x,y)*, we can evaluate their similarities. Here, we define the Gabor response at the point (x_0, y_0) as follows:

$$G_{\theta,\gamma,\sigma}(x_0, y_0) = (f * g_{\theta,\lambda,\sigma})(x_0, y_0) = \int f(x, y) g_{\theta,\gamma,\sigma}(x_0 - x, y_0 - y) dxdy \quad (4)$$

where * represents convolution. By selectively, changing each of the Gabor function parameters, we can tune the filter response (Equation (4)) to particular patterns such as blood vessels arising in the images.

2.3 Feature Extraction

When the RGB components of the retinal images are visualized separately, the green channel represents the best vessel-background contrast. Thus, the green channel was selected to be processed by the Gabor flters. To initially locate the blood vessels from the retinal images, we used a Gabor filter bank $G_{\theta,\gamma,\sigma}$ arranged in 12 orientations (θ spanning from 0^o up to 165^o at steps of 15^o), 3 wavelengths (γ= 1.5, 2.5, 3.5) and 3 scales ($\sigma = 3,5,7$). The wavelength and scale values were experimentally tuned according to our prior knowledge of retinal image characteristics and in such a way to assign stronger responses to pixels associated with the blood vessels of all possible widths. For each considered set of wavelength and scale parameters, we were interested in the Gabor filter response with maximum values over all possible orientations. These values were then taken as the main components of the pixels feature vectors.

Having primarily extracted the candidate blood vessels network based on Gabor filter responses, the image pixels were then classified in terms of vessels and non-vessels. To do that, the pixel feature space was mainly constituted by maximum Gabor filter responses of all orientations taken at 3 different wavelengths, i.e. γ= 1.5, 2.5, 3.5, and 3 scales at σ= 3,5,7 pixels. Indeed, an odd-sized square window was centered on each underlying pixel x_0 in the image. Then the L*uv* color components [10] of the pixels in the window (in an 8-connectivity manner) composed into the feature vector of x_0. There might be no constraint on the neighborhood window size in theory, but it is assumed that most contextual information is presented in a small neighborhood of the x_0 pixel. Her, to determine the optimal window size, we examined various sizes and obtained the best results with a 3x3 window. The total number of features for each pixel was therefore 3x3+9x3=36.

2.4 Pixel-Level Blood Vessels Classification

Here, we analyzed the performance of several classifier models to select the one with the most accurate results. We chose one very commonly used model, for every type of generative and discriminative based approaches, i.e. Gaussian mixture model and support vector machines towards our pixel-level blood vessel recognition task.

2.4.1 Gaussian Mixture Model Classification

GMM classifiers have been utilized in various applications of computer vision and medical imaging [11]. Basically, in a mixture model distribution, the data density is represented as a linear combination of component densities in the form:

$$p(x_i) = \sum_{k=1}^{K} p(x_i \mid w_k; \Theta_k) P(w_k) \tag{5}$$

where K represents the number of components and each component is defined by w_k and parameterised by Θ_k (mean and covariance density parameters). The coefficient $P(w_k)$ is called the mixing parameter. We benefited from the theory behind these models and used two separate mixture of Gaussians to estimate the class densities $p(x|C_i, \Theta)$ of vessels and non-vessels, as follows:

$$p(x \mid C_i, \Theta) = \sum_{k=1}^{K_i} \frac{P(w_k)}{(2\pi)^{\frac{d}{2}} \det(\sum_k)^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(x-\mu_k)^T \sum_k^{-1} (x-\mu_k)\right\} \tag{6}$$

μ_k and $\sum_k$ denote the mean and covariance of the k_{th} component of the mixture density of class C_i. K_i denotes the numbers of components in class i and C_i refers to either the vessel (C_{Vessel}) or non-vessel ($C_{Nonvessel}$) class.

Here, EM algorithm is utilised and its parameters are initialised using a K-means clustering algorithm [12]. In order to apply K-means algorithm, the number of clusters, i.e. K, needs to be known. In this study, the appropriate number of components was chosen by repeating the density model estimation and evaluating a criterion (Minimum Description Length (MDL)) by varying the number of components [13].

2.4.2 Support Vector Machines Classification

Discriminative SVMs have become an increasingly popular tool for machine learning tasks involving classification and regression [14]. For a classification task, the idea is to map the training data into a higher dimensional feature space where a separating hyperplane (w,b), with w the weight vector and b the bias, can be found which maximises the margin from the closest data points. The optimum separating hyperplane can be represented based on kernel function:

$$f(x) = sign(\sum_{i=1}^{n} \alpha_i y_i K(x_i, x) + b) \tag{7}$$

where n is the number of training examples, y_i is the label value of example i, K represents the kernel, and α_i coefficients must be found in a way to maximise a particular Lagrangian representation. However, in real-world problems data are noisy and in general there will be no linear separation in the feature space. The hyperplane margins can be made more relaxed by penalizing the training points the system misclassifies (soft margin). In this case, the learning process of the *SVMs* classifier is equivalent to solving a minimization problem with the objective function of the form:

$$\min_{w \in X} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i \tag{8}$$

The penalty C is a regularisation parameter that controls the trade-off between maximizing the margin and minimizing the training error.

2.5 Experimental Results

The optimum mixture model of each vessel and non-vessel pixels datasets was separately obtained using MDL and by varying the number of components within a range

of 1 to 20. We found, the optimum number of components for vessel and non-vessel pixels 5 and 8 respectively.

The performance of the selected *GMMs* was then quantified based on its sensitivity, specificity and the *overall accuracy* (fraction of correctly classified pixels).The best overall classification accuracy obtained was 95.24%, with 96.14% sensitivity and 94.84% specificity. Alternatively, we classified our manual segmented vessel and non-vessel pixels using *SVMs*. Specifying a *SVM* classifier requires two parameters: the kernel function and the regularisation parameter *C*. Here, the *SVM* classifiers utilized Gaussian radial basis kernel functions [15]. To obtain the optimal value for *C* and the kernel function parameter (σ), we experimented with different *SVMs* using a range of values. In the first experiment, with no restrictions on the Lagrange multipliers (hard margin), we achieved an optimum overall accuracy of 94.45% with 93.42% sensitivity and 95.51% specificity.

In second experiment, to investigate the effect of the soft margins, we evaluated the generalization ability of the different *SVM* classifiers on the training set with σ fixed at 2.5 and for a wide range of *C* values. In this case, the best overall accuracy, using the soft margins technique (referred to as *SVM**) increased to 96.75% with 96.50% sensitivity and 97.10% specificity at *C*=7.0.

In most medical applications the *overall accuracy* may not be a sufficient measure to choose the optimal configuration. Thus, to assess the behavior of the best classifiers throughout a whole range of the output threshold values, ROC [16] curves shown in Figure 3 have also been produced. The bigger the area under the ROC curve (A_z), the higher the probability of making a correct decision. Here, both *GMM* and *SVM** classifiers achieved high performances with areas 0.965 and 0.974 respectively. However, as Figure 2 illustrates, the *GMM* classifier shows to some extent lower performance over the entire ROC space.

So far we have discussed pixel-level classification. We can use our trained classifiers to evaluate the effectiveness of our proposed approach by assessing the whole image pixels. To do that, we considered a population of 20 new retinal images. Each image was then evaluated using the *SVM** classifier and a final decision was made to discriminate the pixels which are belong to the vessel class. Figure 3 shows two typical normal and abnormal images that have been classified at pixel-level.

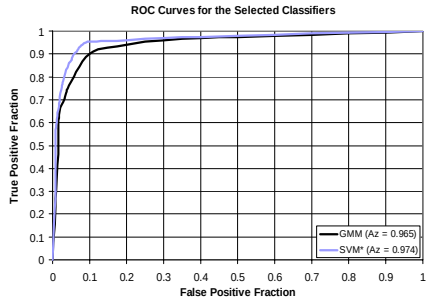
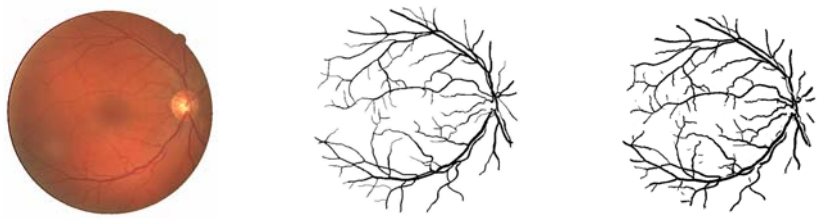
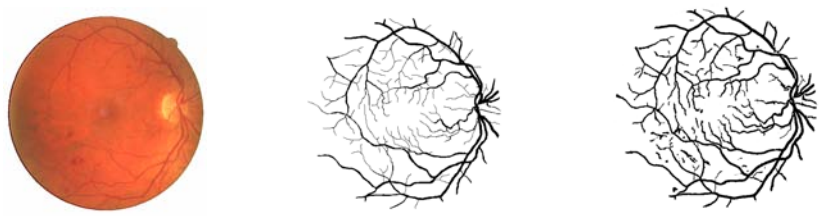


Fig. 2. ROC curves for vessel pixels classification using optimum *GMM* and *SVM* classifiers



(a) Original normal image (b) Manually segmented vessels (c) *SVM** classified vessels



(d) Original abnormal image (e) Manually segmented vessels (f) *SVM** classified vessels

Fig. 3. Retinal blood vessel classification using optimum SVM* classifier

3 Conclusions

The results we have obtained suggest that pixel-level classification in conjunction with Gabor filter responses, feature extraction and *SVM* classifiers can provide robust blood vessel segmentation while suppressing the backgrounds. Although, the majority of large and small vessels are detected, there are some erroneous false detection of noise and other artifacts. The major errors are due to background noise and non-uniform illumination across the retinal images, border of the optic disc and other type of pathologies that present strong contrast. Another difficulty is the lack of precision to capture some of the thinnest vessels that are barely perceived by the human observers. The results by the two classification approaches i.e. *GMM* and *SVMs* are very similar; however, we believe that *SVMs* are a more practical solution to our application as they always converge to the same solution for a given data set regardless of initial conditions, and finally, they remove the danger of over fitting. A perfect medical system would yield an area under ROC curve $A_z = 1$, which is not reached by our method. Alternatively, our experimental results show that the area under the ROC curve reached a value 0.974 which is highly comparable and to some extent higher than the previously reported vessels segmentation methods. Indeed, our method, achieves a sensitivity of 96.50% with a specificity of 97.10%.

References

1. Klein, R., Klein, B., Moss, S., Davis, M., Demets, D.: The Wisconsin epidemiologic study of diabetic retinopathy II. Prevalence and risk of diabetic retinopathy when age at diagnosis is less than 30 years, Archives of Ophthalmology 102, 520–526 (1984)
2. Early Treatment Diabetic Retinopathy Study Research Group, Early Photocoagulation for Diabetic Retinopathy: ETDRS Report 9, Ophthalmology 98, 766–785 (1991)

3. Cree, M.J., Leandro, J.J., Soares, J.V., Jelinek, H.F.: Comparision of Various methods to delineate blood vessels in retinal images. In: Proc. Of the 16thNational Congeress of the Australian Institute of Physics, Canberra, Australia (2005)
4. Chaundhuri, S., Chatterjee, S., Katz, N., Nelson, M., Goldbaum, M.: Detection of Blood Vessels in Retinal Images Using Two-Dimensional Matched Filters. IEEE Trans. on Medical Imaging 8(3), 263–269 (1989)
5. Cote, B., Hart, W., Goldbaum, M., Kube, P., Nelson, M.: Classification of Blood Vessels in Ocular Fundus Imgaes, technical report, Computer Science and Engineering Dept, University of California, San Diago (1994)
6. Tolias, Y., Panas, S.: A Fuzzy Vessel Tracking Algorithm for Retinal Images Based on Fuzzy Clustering. IEEE Trnas. on Medical Imaging 17(2), 263–273 (1998)
7. Staal, J., Abramoff, M., Niemeijer, M., Viergever, M.: Ridge-based vessel segmentation in color images of retina. IEEE Trans. on Medical Imaging 23(4), 501–509 (2004)
8. Osareh, A., Mirmehdi, M., Thomas, B., Markham, R.: Automated Identification of diabetic retinal exudates in digital color images. British Journal of Ophthalmology 87, 1220–1223 (2003)
9. Drimbarean, A., Whelan, P.: Experiments in color texture analysis. Pattern Recognition Letters 22(10), 1161–1167 (2001)
10. Osareh, A.: Automatic identification of diabetic retinal exudates and the optic disc, PhD Thesis, Computer Science Department, Bristol University, UK (January 2004)
11. Rantanen, V., Denessiousk, K., Gyllenberg, M., Koski, T., Jhonson, M.: A fragment library based on Gaussian mixtures predicting favourable molecular interactions. Journal of Molecular Biology 313, 197–214 (2001)
12. Bishop, C.: Neural Networks for Pattern Recognition. Oxford University Press, Oxford (1995)
13. Rissanen, J.: A universal prior for integers and estimation by minimum description length. Annals of Statistics 11, 416–431 (1983)
14. Burges, J.: A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery 2(2), 121–167 (1998)
15. Osareh, A., Mirmehdi, M., Thomas, B., Markham, R.: Classification and Localisation of Diabetic-Related Eye Disease. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2353, pp. 502–516. Springer, Heidelberg (2002)
16. Metz, C.: ROC methodology in radiological imaging. Investigate Radiology 21, 720–733 (1986)

A New Segmentation Method for Iris Recognition Using the Complex Inversion Map and Best-Fitting Curve

Sepehr Attarchi[1], Karim Faez[2], and Mir Hashem Mousavi[3]

[1, 3] Department of Electrical Engineering, AmirKabir University of Technology, Tehran, Iran
{threepehr,mhmm}@aut.ac.ir
[2] Professor of Electrical Engineering Department, AmirKabir University of Technology, Tehran, Iran
kfaez@aut.ac.ir

Abstract. Iris segmentation plays a vital role in automated iris recognition. In this paper, we presented a novel method for iris segmentation using a complex mapping and best-fitting curve procedure. We used an intensity threshold method to extract the rough region of the pupil. For the outer boundary a median filter with prewitt compass edge detector were used to localize the rough region of the outer boundary. By selecting the bottom point of the pupil, which is not usually occluded by the eyelids and eyelashes, as a reference point, two sets of intersecting points between the horizontal lines and pupil's inner and outer boundaries were created. Each point set was map into a new complex domain using the complex inversion map function and the best-fitted line was found on the range. Exact inner and outer boundaries of the iris were found by remapping the best-fitted lines to original domain. We tested our proposed algorithm by implementing a ground truth method. Experimental results show that the proposed method has an encouraging performance.

Keywords: Personal identification, Human iris, Iris segmentation, Complex inversion map, Best-fitting curve, 1D Log-Gabor filter, Hamming distance.

1 Introduction

Biometric personal identification is one of the most interesting topics in recent years. By increasing demands for security in the societies, the old methods of personal identification are not reliable anymore. Iris recognition is the most accurate biometrics which has received increasing attention in departments which require high security. Iris segmentation is an important step in the iris recognition approaches. Iris is usually occluded by top and down eyelids and eyelashes, makes it difficult to extract the iris region accurately. Also reflection spots may cause some problems during the segmentation procedure.

Most of the previous segmentation methods use an integrodifferential operator or Hough transform in order to localize the iris boundary accurately. Daugman [1-3] proposed an integrodifferential operator in order to detect the inner and outer boundary of the iris. Ma et al. [4] used an edge detection method and Hough transform for

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 364–371, 2008.

iris localization. Wildes [5] used Hough transform method and a voting procedure in order to locate the iris boundaries. Although these methods are all accurate and robust to noise, but due to their massive computations, speed of the segmentation in these methods is low and they usually take a lot of time in order to segment the iris boundaries. In the recent years, more studies were carried out to improve the speed of the iris segmentation. Xiaofu He [6] proposed a new segmentation method for hand-held capture devices. They used groups of 3 points and a geometric approach to find the pupil boundary. They used Hough transform to detect the outer boundary of the iris. Li Yu [7] used a new method for iris segmentation in their proposed method. They scanned the binarized iris image horizontally to find the longest chord in the image, assuming the point that the longest chord will be the diameter of the pupil. They calculated the outer boundary parameter using an edge detection method in certain regions.

In this paper, we proposed a new segmentation method using complex inversion map and best-fitting line on the range. We binarized the image using an intensity threshold method. After this, the binarized image was denoised using a morphological operation (opening). In order to detect the boundary of the pupil, canny edge detector was applied to the denoised image. By finding the bottom point of the pupil, a set of intersecting points between the horizontal lines and the pupil boundary was created. By choosing a reference point in the set, all the points in the set were mapped into a new domain, using the complex inversion map. The best-fitting line was calculated in the new domain. By remapping the best-fitting line, the best-fitting circle for inner pupil boundary was found. For the outer boundary, a median filter was applied to the iris image to smooth the iris pattern as much as possible. Then we used a prewitt compass operator [8] in order to detect the outer boundary of the iris. We excluded the pupil region regarding the calculated pupil center and radius and the prior-knowledge that Iris radius has been typically between 90-125 pixels [9]. A set of intersecting points between the horizontal lines and the iris outer boundary was created and the same procedure, as used in pupil boundary detection, was performed to extract the exact region of the outer boundary. Upper part of the iris is usually occluded with top eyelid and eyelashes; therefore the lower part of the iris was used for the recognition approach.

The rest of the paper is organized as follows: Section 2 introduces the pupil boundary detection. Section 3 presents outer boundary detection. Iris normalization, encoding and matching are presented in section 4. Experimental results are given in section 5; finally section 6 concludes this paper.

2 Pupil Boundary Detection

The iris is a thin circular part which lies between the pupil and sclera. Upper and the lower part of the iris are usually occluded by the top and down eyelids and eyelashes. Both the inner and the outer boundary of the iris can be approximately taken as a circle. We used an intensity threshold method to detect the boundary of the pupil considering the point that the pupil is usually darker than its surroundings.

2.1 Pupil Region Extraction

In order to extract the exact region of the pupil, an intensity threshold method was used. Since the pupil is usually darker than the other parts of the image, we binarized the image using a threshold. The threshold was selected using the gray level histogram of the iris image. Fig. 1(a) shows the original iris image and Fig. 1(b) shows iris image after applying the threshold. Upper part of the pupil is usually occluded by top eyelid and eyelashes which their intensity is similar to pupil. On the contrary the lower part of pupil is rarely affected by any eyelids and eyelashes. In order to remove the noisy effect of eyelids and eyelashes in the binarized image, we used a morphological operation (opening). First we applied a disk operator to erode the image, and then the same disk was used to dilate the eroded image. Fig. 1(c) shows the result of applying the morphological operator. For extracting the rough pupil region in the iris image, we used a Canny edge detector to detect the edges of the pupil as shown in Fig 1. (d). As we mentioned before, bottom point of pupil is robust to noise and usually not occluded by the top and bottom eyelids and eyelashes. In addition, lower eyelid and eyelashes intensities are greater than pupil intensity [9]. Therefore they can be completely removed by applying the threshold. We scanned the image horizontally in order to locate the bottom point of the pupil. By finding the bottom point, we scanned the pupil horizontally from the bottom upward at regular intervals and created a set of intersecting points between the horizontal lines and pupil boundary. In the next step, a complex inversion map is used to estimate the best fitting circle of the pupil boundary. Complex inversion map is defined as:

$$f(z)=\frac{1}{z} \qquad z=x+iy \tag{1}$$

This function maps a circle on the domain which passes through the origin, to a straight line on the range. Regarding this, 5 steps are performed to get the best fitting circle for inner boundary:

1. Rearranging the point set by converting each pair (x, y) to a complex number $x+iy$.
2. Randomly selecting a reference point from the reshaped point set.
3. Subtracting the coordinates of the reference point from the other points in the set.
4. Mapping each point in the point set using the complex inversion map. *u* and *v* in the new domain are calculated from the equations below:

$$u=\frac{x}{x^2+y^2} \qquad v=\frac{-y}{x^2+y^2} \tag{2}$$

5. Finding the best fitting straight line through the mapped points in the new domain. With the prior-knowledge that the vertical offsets from a line are almost always minimized instead of the perpendicular offset, the best fitting line with the parametric equation;

$$f(x, y)=ax+b \tag{3}$$

Which passes through *n* different points is calculated as below:

$$a = \frac{\sum_{i=1}^{n} y_i \sum_{i=1}^{n} x_i^2 - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} x_i y_i}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i^2\right)^2}$$
$$b = \frac{n\sum_{i=1}^{n} y_i x_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i^2\right)^2} \quad (4)$$

6. Remapping the calculated equation to obtain the best fitted circle for pupil boundary.

Fig 1. (e) shows the horizontal lines which intersect the pupil boundary and Fig 1. (f) illustrates the extracted pupil boundary.

3 Outer Boundary Detection

Detecting the outer boundary of the iris is more complicated than the inner boundary because of the little contrast between the iris and the sclera. In order to detect the outer boundary, we used 6 major steps as mentioned below:

1. Applying a 13×13 median filter to the iris image, mainly to smooth or completely remove the iris pattern. Fig 2. (b) shows the filtered image.
2. Convolving the left side of the image by 0° oriented prewitt compass edge detector and the right side by 180° oriented prewitt compass edge detector to detect the outer boundary edges.
3. Using an intensity threshold method in order to exclude the weak detected edges. Result is shown in Fig 2. (c).

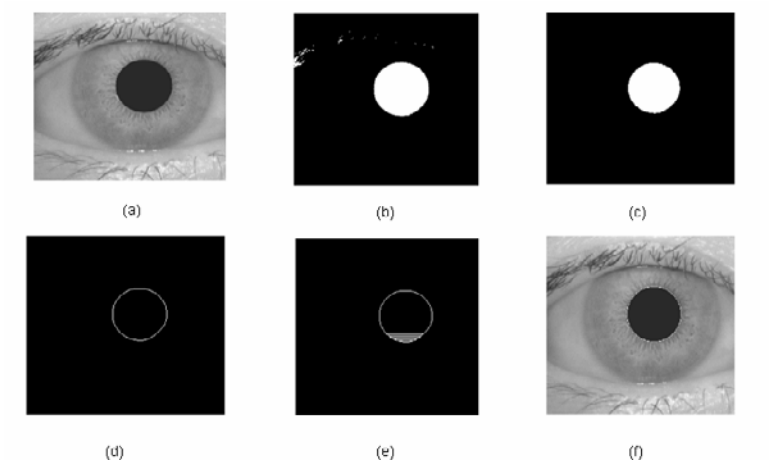


Fig. 1. Pupil localization. (a) Iris image. (b) Binarized image. (c) Denoised image after applying morphological operator. (d) Edge image. (e) Horizontal lines intersecting pupil boundary. (f) Detected pupil boundary.

4. Excluding the pupil region as shown in Fig 2. (d), regarding the calculated pupil center and radius in the previous section and the prior-knowledge that Iris radius has typically been 90-125 pixels [9].
5. Scanning the image horizontally from the bottom of the pupil upward and downward at regular intervals and created a set of the intersecting points between the horizontal lines and outer iris boundary. Fig 2. (e) shows the horizontal lines which intersect the outer iris boundary.
6. Performing all the steps mentioned in section 2.1 to find the exact outer boundary of the iris.

4 Normalization, Encoding and Matching

Different people have different iris sizes. Even the size of the iris captured from one person may be different due to illumination variations. In order to solve this problem we used daugman's rubber sheet model to unwrap the iris into the rectangular block of the fixed size 12×240. Upper part of the iris pattern is usually occluded by the top eyelid and eyelashes. In order to avoid these noisy effects in the extracted iris pattern, we used only the lower half of the iris. Therefore we excluded the first 120 columns of the normalized iris images making our extracted template of the size 12×120. Applying the 1D-Log-Gabor filter results in 12×120 complex iris pattern. Phase quantization was performed to encode the complex pattern into a simple binary code. We classified the angles to four different classes and the binary codes 11, 01, 00, and 10 were used to represent four classes respectively. In the matching step, Hamming distance was used as a measure of the similarity.

5 Experimental Results

In order to evaluate the effectiveness of our proposed algorithm, extensive experiments were performed. We designed and implemented a semi-manual ground truth method as Xiaofu He used in [6] to evaluate the accuracy of our proposed segmentation method. Further we tested our iris recognition approach in both identification and verification mode. We used the public database CASIA [10] which includes 756 iris images from 108 different people. Each person has 7 different iris images captured in 2 sessions with 320×280 pixels in 256 gray levels.

5.1 Segmentation Results

We tested our proposed segmentation method by implementing a semi-manual ground truth method similar to what Xiaofu He et.al. used during their experiments [6]. We used the same idea to evaluate the performance of our segmentation results with a little change due to differences between our proposed method and theirs. Following steps were performed in order to generate the ground truth:

1. Extracting the pupil region using the same method in section 2.1.
2. Detecting the edge of the pupil as described in section 2.1, also deleting the noises in the image manually.

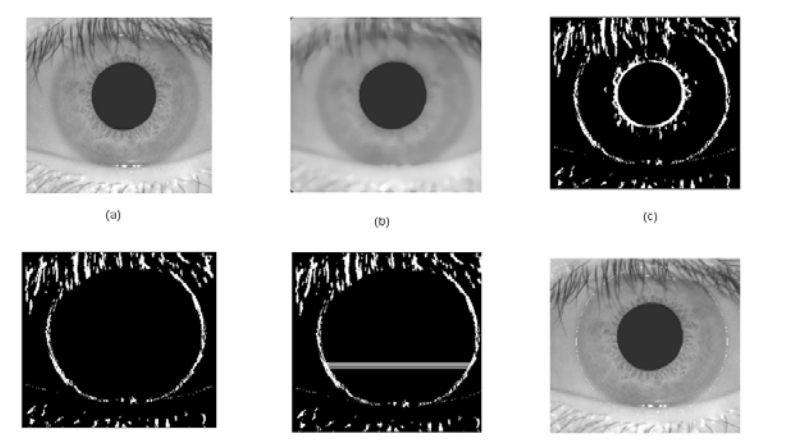


Fig. 2. Outer boundary detection. (a) Iris image. (b) Iris image after applying median filter. (c) Iris image after applying Prewitt compass edge detector. (d) Edge image after excluding the pupil boundary. (e) Horizontal lines intersecting iris outer boundary. (f) Detected iris outer boundary.

3. Calculating the pupil center and radius using Hough transform.
4. Detecting the edge of the outer boundary of the iris using the method described in section 3.
5. Excluding the pupil region and deleting the noises in the image manually.
6. Calculating the outer boundary circle parameter using the Hough transform.

After performing the ground truth method, we compared the result of the segmentation obtained by this method with our proposed method. We calculated the distance between the center and radius of the inner and outer boundary circles in both ground truth and our proposed method, considering the point that the smaller distance results in a better segmentation. If the distance between our proposed method and the ground truth method for pupil center and radius was within the 8 pixels, we considered this as a success. Otherwise the result is considered as a failure. For the outer boundary

Table 1. Comparison results of the inner and outer boundary parameter

Methods	Pupil center distance (pixels)		Pupil Radius distance (pixels)		Accuracy (%)	
	Mean	Standard Deviation	Mean	Standard Deviation	Center	Radius
Proposed	1.24	1.14	1.12	1.02	99.86	99.95
Xiaofu He method	2.20	1.43	1.44	1.24	99.33	99.92
	Outer boundary center distance (pixels)		Outer boundary radius distance (pixels)		Accuracy (%)	
Proposed	3.26	2.63	1.34	1.87	98.78	99.55
Xiaofu He method	4.2009	3.7525	1.3858	2.9076	98.0	99.42

the distance threshold was selected 15 pixels as used by Xiaofu He in [6]. Table 1 shows the segmentation results of the inner and the outer boundary. Also the Xiaofu He's results were mentioned in the table. As it is clear, our algorithm has better segmentation results in both accuracy and speed. All the calculation was implemented using Matlab 7 on an Intel Pentium IV 2.66G processor with 512 MB memory, whereas Xiaofu He et.al. implemented their proposed method using Matlab7 on an Intel Pentium IV 3G processor with 512 MB memory. The execution time for inner boundary detection was 0.16s and 0.055s for outer boundary detection. Table 2 shows the execution time for the inner and the outer boundary detection.

Table 2. Time comparison results

Method	Inner boundary detection time(s)	Outer Boundary detection time(s)
Proposed	0.16	0.055
Xiaofu He method	0.18	0.56

5.2 Recognition Results

Proposed recognition method was tested in both identification and verification mode. The first 3 iris images of each person (session 1 in CASIA database) were chosen to build a template and the other iris images for testing. Our proposed algorithm achieved the correct recognition rate (CCR) of 98.2%.

6 Conclusion

In this paper, we proposed a new segmentation method using the complex inversion map and the best-fitting line on the new complex domain. We extracted the pupil region using an intensity threshold method and then the binarized image was denoised using a morphological operation. Canny edge detector was used to detect the edge of the pupil boundary. By finding the bottom point of the pupil, a set of intersecting points between the horizontal lines and the pupil boundary was created. After choosing a reference point from the set, all the points in the set were mapped on the range using the complex inversion map. Best fitting line was calculated in the new domain and by remapping the line, the best fitting circle for inner pupil boundary was found. For the outer boundary, first we used a median filter to smooth the iris pattern as much as possible. Then we used a prewitt compass operator in order to detect the outer boundary of the iris. We applied a 1D-Log-Gabor filter to each row of the 2D normalized iris pattern and then we quantized the 2D complex pattern into four different classes considering their phase. At last the Hamming distance was used as a measure of similarity. We implemented a ground truth method in order to evaluate the accuracy of our proposed segmentation algorithm. We reached the correct segmentation rate of 99.86, 99.95, 98.78 and 99.55 for the pupil center, pupil radius, outer boundary circle center and outer boundary radius respectively. The average execution time for inner boundary detection was 0.16s and 0.055s for outer boundary detection

using Matlab 7 on an Intel Pentium IV 2.66G processor with 512 MB memory. Further we calculated the recognition results. Our algorithm reached the correct recognition rate of 98.2%.

References

1. Daugman, J.G.: High confidence visual recognition of person by a test of statistical independence. IEEE Trans. Pattern Anal. Mach. Intell. 15(11), 1148–1160 (1993)
2. Daugman, J.G.: The importance of being random: statistical principles of iris recognition. Pattern Recognition 36(2), 279–291 (2003)
3. Daugman, J.G.: How iris recognition works. IEEE Trans. Circuits Syst. Video Technol. 14(1), 21–30 (2004)
4. Huang, J., Wang, Y., Tan, T., et al.: A new iris segmentation method for recognition. In: Proceedings of the 7th International Conference on Pattern Recognition, vol. 3, pp. 554–557 (2004)
5. Wildes, R.: Iris recognition: an emerging biometric technology. Proc. IEEE 85(9), 1348–1363 (1997)
6. He, X., Shi, P.: A New segmentation approach for iris recognition based on hand-held capture device. Pattern Recognition 40, 1326–1333 (2007)
7. Yu, L., Zhang, D., Wang, K.: The relative distance of key point based iris recognition. Pattern Recognition 40, 423–430 (2007)
8. http://www.cee.hw.ac.uk/hipr/html/prewitt.html
9. Al-Zubi, R.T., Abu-Al-nadi, D.I.: Automated personal identification system based on human iris analysis. Pattern Anal. Applic. 10, 147–164 (2007)
10. CASIA, CASIA iris image database (2003) http://www.sino-biometerics.com

A New Algorithm for Combating Free-Riders in Structured P2P Networks*

Mohammad R. Raeesi N.[1], Jafar Habibi[2], Pariya Raoufi[3], and Habib Rostami[4]

Computer Engineering Department, Sharif University of Technology,
Tehran, Iran
raeesi@ce.sharif.edu
habibi@sharif.ir
raoufi@ce.sharif.edu
habib@sharif.ir

Abstract. The strength of a peer-to-peer (P2P) network depends on collaboration of participant nodes in file sharing. Free-riders are nodes that join the network and download files without sharing any files to be downloaded by others. In this paper, we introduce a novel algorithm to motivate nodes of structured P2P networks to share files and to limit downloads of free-riders. We measure effectiveness and efficiency of the algorithm using simulation and show that the algorithm significantly reduces percent of downloads of free-riders.

Keywords: Peer-to-Peer Network, Free-Rider, Incentives, Collaboration.

1 Introduction

In the P2P networks, there is not any central control on the peers. They are self-organizing; so there is no control on contribution of the peers, while P2P systems rely on contribution of their peers. Peers serve files free for other nodes, so they do not benefit from serving files to the others. However, users want to maximize their benefits from the system and it may conflicts with systems goal and performance.

The nodes on the network that share no files and only use other nodes' resources are called free-riders. Studies on P2P networks show that they suffer from large number of free-riders. Nevertheless, reports show that many P2P systems persist in spite of this large number. An extensive analysis on Gnutella [1] shows that 70% of Gnutella users are free-riders. This analysis also demonstrates that free-riders are distributed evenly between domains, so that no one group contributes significantly more than others do [2].

Free-riders decrease the network performance and have more vulnerability for the system. When the number of free-riders in the network increases, system will lose its growing rate, and it may be collapsed. There is some ways to motivate nodes to collaborate on the network.

* This research was in part supported by a grant from Iran Telecommunication Research Center.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 372–379, 2008.

The result of this paper is organized as follows: In section 2, first, we survey on the related work, and then in section 3, we propose a new algorithm for combating free-riders. Free-riders need to cooperate for using the network resources. Each node has credit for downloads, but does not need to pay for its downloads, because its uploads will increase it. If a node has not sufficient credit for downloads, first it must upload.

At last, we show the effects and results of our algorithm on the P2P networks by simulation of it.

2 Related Work

All ways for combating free-riders are divided to three categories; first one is imperfect that users can cheat the system, second one is expensive that needs servers, and the last one incur high transactional costs [3].

In some file sharing networks such as eDonkey [4], eMule [5], and Pruna [6], are uploading and downloading limits, but users can alter their applications to use the network more than they have been allowed. We can use servers to keep information about nodes' history, but this way has more financial costs. In addition, we can keep these information on the own node. In this way, nodes could change their information.

There are some other method to prevent free-riders from downloading, such as payment scheme in which each downloader should pay the downloading fees [7], and public key infrastructure to add an economic system to the network [3].

Three types of schemes exist for addressing free-riders [8]. These consist of Inherent Generosity (each peer contribute based on its generosity [9]), Monetary Payment and Reciprocity-Based Scheme (each peer can download if it has uploaded).

Some incentives exist to motivate nodes to contribute on the P2P networks. For example, we can consider bandwidth and TTL tag as incentives [10]. We can ask uploaders to divide their uploading bandwidth between downloaders based on their contribution. In addition, we can set the TTL tag based on this; if a node is free-rider we can decrease it, so the percent of the node's responded requests will be decreased.

3 Proposed Algorithm

In this section, we present our proposed algorithm. The algorithm addresses free-riders and does not allow them to continue their work on the network. This algorithm defines credit for each node and saves credit information distributively. Our algorithm is based on DHT, so it needs a DHT-based overlay network (Our algorithm can work on every DHT-based overlay networks.). We design our algorithm for decentralized, structured P2P networks. By this algorithm, we allow a node to download as much as it uploads. At first, we consider an initial credit for each node to download a few files.

In our algorithm, we save two types of information distributively based on DHT. These are information of uploaders, which is kept on the nodes that are responsible for the result of hashing the files' specification, and the credit information of nodes, which is kept on the nodes that are responsible for the result of hashing the nodes ID. In this paper, we call the responsible of each node as its indexer.

The design of the proposed algorithm is based on some assumptions, which we review in subsection 3.1. In subsection 3.2, we describe messages, which are needed to convey for updating the credit information.

3.1 Algorithm Assumptions

The followings are our design assumptions:

1. The goal of this algorithm is addressing free-riders and preventing them from unlimited downloading.
2. Goal of free-riders is downloading more and more. Therefore, they do anything, which allows them to download more. They do not anything only to hurt others.
3. We do not consider the correctness of the files on the network and leave it to trust algorithms.

3.2 Algorithm Messages

Like some common algorithms, downloader requests to download a file. It hashes file's specification to find a node, which keeps the information of the file, and send its request to it. The node sends the information of file's uploaders to the downloader as a response. After selecting an uploader, downloader sends its request to it.

Our algorithm is different from others, after uploader gets downloader's request. We have the sequence of messages for managing the credits of downloader and uploader. This sequence is:

1. Downloader requests uploader to download a file.
2. Uploader sends credit-checking message to the downloader-indexer. This message has the downloader's ID and file's specification.
3. Downloader-indexer checks the credit of downloader. If the credit is not enough for downloading this file, it sends a message to uploader and says that downloader have not permission for downloading, and then uploader says to downloader about the matter. Otherwise, if the credit is enough for downloading that file, download-indexer decreases the credit of downloader, sends a message to uploader-indexer, and tells it.
4. Uploader-indexer says to uploader to upload the file.
5. For checking the correctness of node's claim to be the uploader-indexer, uploader sends a message to its real indexer.
6. Uploader-indexer replies this message.
7. Uploader uploads the file for downloader.
8. When download is complete, downloader tells both downloader-indexer and uploader-indexer about that. Then uploader-indexer increases the credit of the uploader.
9. If in downloading, error occurs, downloader tells this occurrence to the uploader-indexer.
10. Uploader-indexer checks this claim by asking confirmation of uploader.
11. Uploader replies uploader-indexer.
12. If uploader confirms the error's occurrence, uploader-indexer sends a message to downloader-indexer to increase the decreased credit of downloader. Nevertheless, if uploader does not confirm that, neither the credit of downloader will be increased, nor the credit of uploader.

4 Experimental Result

We use PlanetSim 3.0 [11] for the simulation. This is an object-oriented simulator that has been developed using JAVA language programming. This simulator is used for structured P2P networks and DHT-based services for overlay networks.

To do the experiments, first, we generate specific number of nodes, and then generate specific number of files and copying from each one different number using Zipf distribution. Then, we select randomly (uniform distribution) from nodes for loading generated files. At last, loaders send their files' information to identifiers, which are specified by hashing the files' specification, as a message. Then each node decides to do something, including leaving network, joining it and requesting a file.

In this simulation, we consider that a live node decides to leave the network with 10% probability, to request a file with 50% probability, and to do nothing with remained probability. In addition, we consider that a leaved node decides to join the network with 90% probability, and to remain outside the network with 10% probability. In the simulation, we let nodes to decide and work in 20 cycles based on the considered probabilities. For showing the nodes' operations, we execute the algorithm in 50 cycles.

Another important matter is the initial credit, which is given to each node for being able to work in the network. The needed credits for downloading different files are considered as the same. In addition, we consider the initial credit for each node to be equal to the needed credit for downloading two files. Nodes are divided to contributed nodes and free-riders. Each of them has some successful and unsuccessful downloads. Unsuccessful ones are cases which are not done because of the insufficient credits.

5 Analysis of the Results

In this section, we analysis simulation results.

5.1 Effectiveness of Algorithm

Fig. 1 (a) shows that with the maintained configuration, in the absence of our algorithm, there is no meaningful difference between cooperatives and free-riders in terms of successful downloads. Therefore, this is dangerous for networks like Gnutella because it contains 70% free-riders and it causes 70% download traffic for free-riders.

When we employee the algorithm, the rate of free-riders' successful downloads will be decreased significantly. Fig. 1 (b) shows that the number of successful downloads in the network, unlike the case in which we do not use the algorithm, will not be divided equally between contributors and free-riders and the number of contributors' successful downloads are three times more than the number of free-riders' downloads.

Algorithm in long time bans free riders from downloading. This time, algorithm is simulated for networks with 1000 nodes and we continue the simulation with 50 cycles instead of 20 ones. Results of this simulation have been shown in the Fig. 2. In this figure, we compute the percent of successful free-riders' download of the whole downloads.

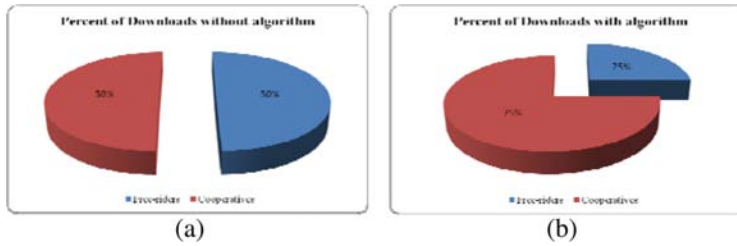


Fig. 1. (a) Percent of downloads without algorithm for cooperatives and free-riders. (b) Percent of downloads with algorithm for cooperatives and free-riders.

Fig. 2 shows that the rate of the successful free-riders' downloads will be decreased as the time passes. It is because, the credits of free-riders will be finished and they do not have permission for downloading other files. This figure is another reason, which shows the suggested algorithm continues our goals.

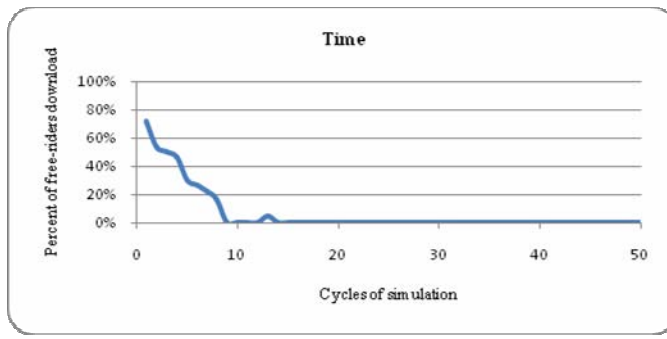


Fig. 2. Result of the proceeding of using algorithm on time

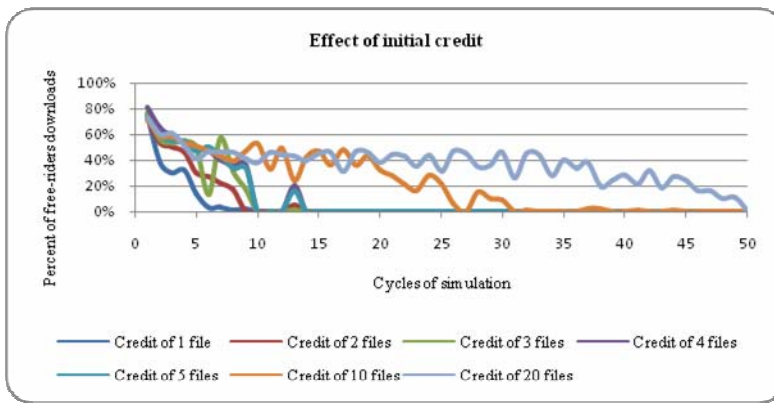


Fig. 3. Effect of the amount of initial credit on the algorithm performance

Off course, the initial credit is important for the time in which the algorithm will be stable. It means that if the initial credit is more than the needed credit for two downloads, we predict that this algorithm needs more time to be stable. This prediction has been proven with simulation. Based on the Fig. 3, we can say for every initial credit, after a time which is proportional to this value, the percent of successful free-riders' downloads reach to zero percentage. Free-riders can use the network in initiation parts of their life cycle and the duration of these parts depends on the initial credit, which a node receives when it joins the network for the first time.

Fig. 4 shows that if we keep the percent of free-riders in the network as a constant, the size of the network or number of nodes in the network has not any effect on the performance of our algorithm. This means that the presented algorithm is scalable and independent of the network size; hence, the algorithm can be deployed in real large-scale P2P networks.

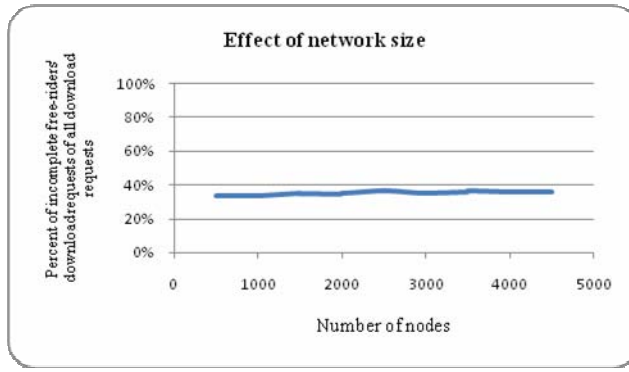


Fig. 4. Effect of network size or the number of nodes in the network on the performance

5.2 Algorithm Efficiency

In terms of efficiency, the algorithm has some traffic overhead, because some message-passing between downloader, uploader and their indexer nodes should be done. Fig. 5 shows that the overhead, which the algorithm imposes to the network, is the same as its basic traffic. Therefore, the network traffic become twice, when we run our proposed algorithm. Nevertheless, we should note that this is not the full story. When we ban free riders, they do not query the network anymore and this causes traffic reduction.

5.3 Effect of Different Factors on the Algorithm

Unlike network size, the percent of free-riders in the network has effect on the algorithm. Fig. 6 shows that if the percent of free-riders in the network is increased, network performance will be increased and we can prevent from more unsuitable downloads. Because increasing in the number of free-riders causes increasing in the number of downloading requests and because they do not have enough credits for downloads, these requests have no reply and will added to unsuccessful downloads.

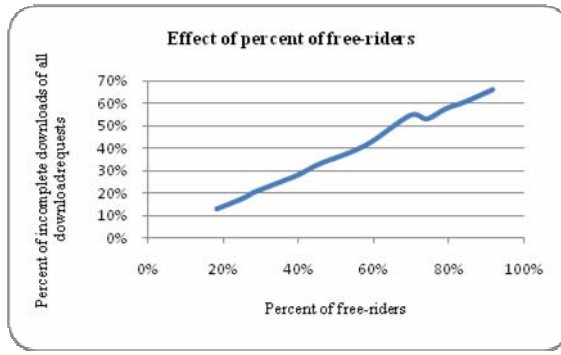


Fig. 5. Effect of the percent of free-rider in the network on the performance of algorithm

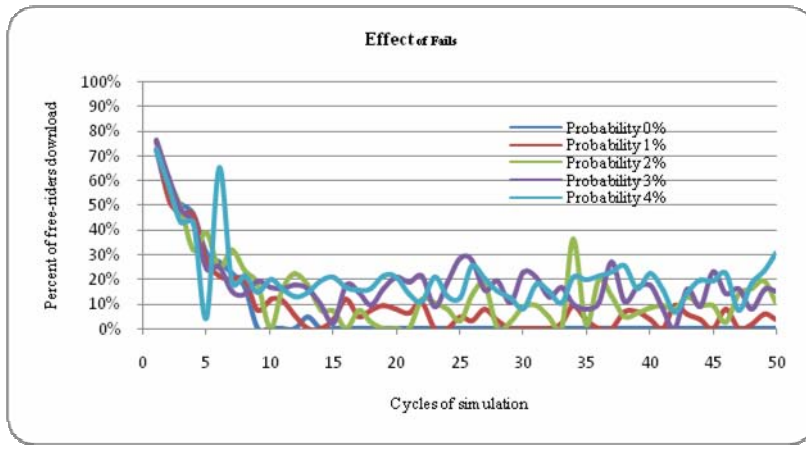


Fig. 6. Effect of failure on the performance of algorithm

Another factor that has important effect on this algorithm is failures of the nodes in the network. If a node wants to leave the network, it gives its information, which it was responsible for it, to another node and then leaves the network. In the failure, a node leaves the network without giving its information to another node and this causes losing of its information.

In the suggested algorithm, files' information and credits' information are kept distributively. In failure, files' information will be constructed again with the mechanisms of DHT-based networks. However, for each node, if its indexer fails, whole of its contribution information will be lost. After this, like when this node enters the network for the first time, its information will be considered as a new node.

For studying the effect of this factor, we allocate a portion of 40% left probability to failure of each node. We have studied this with different probabilities, form 0% to 4%. Fig. 7 shows that if the probability of failure in the network is low then it has no danger for the algorithm. For instance if the probability of failure is less than or equal to 3% then the algorithm will decrease at least 40% of free-riders' downloads.

6 Conclusion and Future Works

A new algorithm for combating free-riders in structured P2P networks was introduced that could effectively ban free-riders from unlimited downloads and motivate them to share files that should be downloaded by others. In addition, we presented effects of different parameters of a P2P network on performance of it. As future work, we are going to generalize the algorithm and deploy it on other types of P2P networks.

References

1. Gnutella website, http://www.gnutella.com
2. Adar, E., Huberman, B.A.: Free riding on Gnutella, Technical Report. Xerox PARC 5(10), Augsut 10 (2000)
3. MyungJoo, H., Agha, G.: ARA: A Robust Audit to Prevent Free-Riding in P2P Networks. In: The Fifth IEEE International Conference on Peer-to-Peer Computing (P2P 2005), pp. 125–132 (2005)
4. eDonkey website, http://www.edonkey2000.com
5. eMule website, http://www.emule-project.net
6. Pruna website, http://www.pruna.com
7. Lai, K., Feldman, M., Stoica, I., Chuang, J.: Incentives for Cooperation in Peer-to-Peer Networks. In: Proceedings of the Workshop on Economics of Peer-to-Peer Systems, Berkeley, California, USA, pp. 1–6 (June 2003)
8. Feldman, M., Chuang, J.: Overcoming Free-Riding Behavior in Peer-to-Peer Systems. ACM Sigecom Exchanges 5(4) (July 2005)
9. Feldman, M., Papadimitriou, C., Stoica, I., Chuang, J.: Free-Riding and Whitewashing in Peer-to-Peer Systems. In: Proc. SIGCOMM workshop on Practice and Theory of Incentives and Game Theory in Networked Systems, pp. 228–236 (2004)
10. Kamvar, D.S., Schlosser, T.M., Garcia-Molina, H.: Incentives for Combatting Freeriding on P2P Networks. In: Kosch, H., Böszörményi, L., Hellwagner, H. (eds.) Euro-Par 2003. LNCS, vol. 2790, pp. 1273–1279. Springer, Heidelberg (2003)
11. PlanetSim website, http://www.planetsim.net

Clustering Search Engine Log for Query Recommendation

Mehdi Hosseini and Hassan Abolhassani

Web Intelligence Research Laboratory, Computer Engineering Department,
Sharif University of Technology,
Tehran, Iran
me_hosseini@ce.sharif.edu, abolhassani@sharif.edu

Abstract. As web contents grow, the importance of search engines became more critical and at the same time user satisfaction decreased. Query recommendation is a new approach to improve search results in web. In this paper we represent a method to help search engine users in attaining required information. Such facility could be provided by offering some queries associated with queries submitted by users in order to direct them toward their target. At first, all previous query contained in a query log should be clustered, therefore, all queries that are semantically similar will be detected. Then all queries that are similar to user's queries are ranked according to a relevance criterion. The method has been evaluated using a real world data set and by comparing it to existing approaches, the results show promising improvements.

1 Introduction

As web contents grow, the significance of search engines turned out to be more essential and simultaneously user's satisfaction decreased. To improve search results, query log analysis is applied to better find out the users' search demands. Each search engine has a repository of query log which can be used to facilitate understanding of users' needs. Query log clustering is required on the way to prepare web queries for such services. Totally all services that exploit query log could be categorized as following approaches:

- • **Improvement of search results:** when a query is submitted, the results of queries in the same cluster with the entered query that are clicked through can be used to improve search results.
- • **Query expansion:** as the researches shows that average query terms are near two [5]. So most of the time, queries are ambiguous. One possible remedy for this problem is to expand a query with new terms. Query clustering helps to find relevant terms for this expansion which can be applied in the two ways: (1) Query expansion in terms of similar queries. (2) Expansion in the terms of selected pages of similar queries.
- • **Personalization of search results:** here the goal is to post process the search results and display different sets to different users based on their attributes. By clustering user queries one can find a user's interests and demands. In a profile for each user is created based on the past queries entered by them.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 380–387, 2008.

– • **Query recommendation:** this method helps users to submit appropriate query to a search engine. With query clustering it is possible to find such appropriate queries. For example, reports a method for this purpose. This strategy is different from expanding queries terms because the expansion method construct artificial query, while the query recommendation gives actual related queries formulated by other users that had the same information in the past. Characteristically, the catalog of recommended queries is computed by processing the query log, which saves the all previously submitted queries and the URL's selected in their answers. A main problem that arises in this context is how to model the information needs associated to a query. Some models proposed in [3] represent a query as the set of URL's clicked by users for the query. This approach is highly sensitive to user noisy selections. The experiments have shown that users' selections are not always true, and sometime some URLs may be preferred, however they are not relevant to a query.

The method presented in this paper is a type of fourth approach, query recommendation. To do this, web queries from a query log are firstly clustered in a reduced dimension space, and then in favor of each query the recommendation process is accomplished regarding to associative clusters.

1.1 Contributions

In this research work an algorithm will be presented in order to suggest related queries to a query submitted by a search engine user. Collections of analogous queries are found by running a clustering process over the queries and their associated URLs in the query logs. The clustering process is based on our previous work [7], relation between queries and clicked URLs on query log which is considered as a bipartite graph, the nodes of one side for queries and the nodes of the other side for clicked URLs. Semantically similar queries may not share query-terms, but same URLs may be clicked by users. Thus our framework avoids the problems of comparing and clustering sparse collection of query-term vectors, in which semantically similar queries are difficult to find, a problem that appears in previous work on query clustering. Further, in our method, concurrently URLs are clustered with queries, so they will be useful for URL recommendation. We provide relevance criterions to rank the suggested queries. Finally, we present an experimental evaluation of the algorithm, using logs from the AOL search engine.

1.2 Related Works

In order to improve different aspects of search engines performance, in [1] a survey on the use of web query log is provided. In [6] a method has been proposed to clustering similar queries in the direction of recommending URLs to frequently asked queries of a search engine. Four notions of query distance are employed: (1) derived from keywords or phrases of the query; (2) based on string similarity; (3) according to shared clicked URLs; and (4) according to the distance of the clicked documents' URLs in some pre-defined hierarchy. In [3] a query clustering technique under distance notion (3) is proposed. In another work, [4], a method to discover related

queries based on association rules is demonstrated. Here queries represent items in traditional association rules. The query log is considered as a collection of transactions, where each transaction shows a *session* in which a user puts forward a sequence of associated queries in a time interval. The method shows high-quality results, however two troubles arise. First, it is difficult to determine sessions of consecutive queries that belong to the same search progression; on the other hand, the most interesting related queries, those submitted by different users, cannot be discovered. This is because the support of a rule increases only if its queries appear in the same query session, and thus they must be submitted by the same user. In [8], a method to recommend queries according to seven different notions of query similarity is proposed. Three of them are moderate variations of notion (1) and (3). Another approach adopted by search engines to suggest related queries is *query expansion* [2]. The idea here is to reformulate the query such that it gets closer to the term-weight vector space of the documents the user is looking for. Our approach as Query recommendation is different, since we study the problem of suggesting related queries issued by other users and query expansion methods construct artificial queries. In addition, our method may recommend queries that are related to the input query but may search for different issues, thus redirecting the search process to related information of interest to previous users. The remainder of this paper is organized as follows. Section 2 describes the query clustering process. In section 3 we present the method proposed for computing and ranking related queries. Section 4 presents the experimental evaluation of the method. Finally, in section 5 we conclude and outline some prospects for future work.

2 Query Clustering

In order to compute the similarity of two queries, we first consider queries and URLs as a bipartite graph, the nodes of one side for queries and the nodes on the other side for clicked URLs. Fig.1, for example, shows a sample of such graph when the left nodes denote queries and the right nodes denote URLs. We symbolize a graph by *G(V,E)*, where *V* is the vertex set and *E* is the edge set of the graph. The graph *G(V,E)* is a bipartite graph among two vertex classes such as *Q* and *U* while $Q \cup L=V$ with $Q \cap L=\emptyset$ and each edge in *E* has one endpoint in *Q* and one endpoint in *U*. We consider

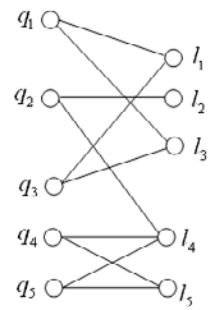


Fig. 1. Query-URL collection as a bipartite graph

weighted bipartite graph *G(Q,L,W)* where *W* is a relation matrix and $w_{ij}>0$ denotes the weight of the edge between vertex *i* and *j*. If there is no edge between them then $w_{ij} = 0$. In query-URLs co-clustering, w_{ij} is used to denote the number of times that URL *j* has been clicked by users who issued the query *i*.

Furthermore, each query is represented as a vector where i^{th} element indicates the relation between the query and URL *i*. A query vector is shown in Equation (1) which $rel(q_k,l_i)$ indicates the value of relationship between query *k* with URL *i*.

$$\overrightarrow{q_k} = (rel(q_k, l_1), rel(q_k, l_2), \dots, rel(q_k, l_m) \tag{1}$$

In this paper such vectors are used to compute similarity between queries and to do clustering. In continue, considering query-URL bipartite graph, we propose a relationship functions which can be used to compute the relation between queries and URLs indicated in the equation (1) as $rel(q_i, l_j)$. Defining relationship function as follow, we consider the total numbers of unique queries equals *n* and the total numbers of unique URLs equals to *m*. In this function, $rel(q_i,l_j)$ is computed by multiplying two parts. The first part is the ratio w_{ij} to the total number of selections of j^{th} URL. The second section is the natural logarithm of the ratio of the total number of unique URLs as $|L|$ to the number of URLs which selected by i^{th} query. Equation (2) shows this computation which connect (q_i,l_k) is a Boolean function (Equation (3)).

$$rel(q_i, l_j) = \frac{w_{ij}}{\sum_{k=1}^{n} w_{kj}} \times \log\left(\frac{|L|}{\sum_{k=1}^{m} connect(q_i, l_k)}\right) \tag{2}$$

$$connect(q_i, l_k)\begin{cases}1 \ ; \ w_{ik} > 0\\ 0 \ ; \ w_{ik} = 0\end{cases} \tag{3}$$

Computing clusters is similar to [7]. In what follow this clustering method is discussed briefly. Firstly, all connected components from the bipartite graph are extracted. Fig. 2 demonstrates two connected components extracted from a bipartite graph depicted in Fig. 1.

For each connected component that contains a huge number of queries, a relation matrix is created. Each row in a relation matrix as *A* corresponds to a query of the related connected component and each column corresponds to a URL in the connected component, so each A_{ij} maintains $rel(q_i,l_j)$. Subsequently, we use singular value decomposition (SVD) [10] to project relation matrix of queries and URLs into reduced dimensions space, therefore noisy connections inside a connected component are eliminated. Dimensions of new space are created by singulars vectors corresponding to largest singular values of matrixes issued by applying SVD. After applying SVD we use Frobenius norm [9] to find the appropriate number of dimensions in new space. In reduced dimension space, the similarity matrix of each connected component is forwarded to K-Means clustering [11] to categorize queries and URLs separately. We use K-Means because of its simplicity and its appropriateness for documents clustering.

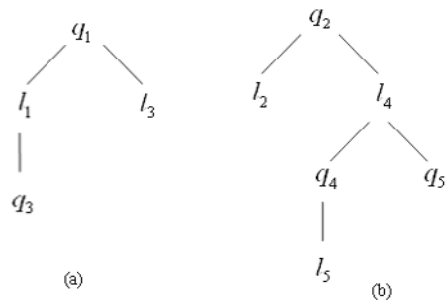


Fig. 2. Two connected components extracted from the bipartite graph in Fig. 1

3 Recommending Helpful Queries

The query recommender algorithm operates in the following steps:

1. Queries along their clicked URL's extracted from the query log are clustered. This is a preprocessing phase of the algorithm, explained in section 2, can be conducted at periodical and regular intervals.
2. At the searching time, given an *input query*, we first find the cluster to which the input query belongs. Then a rank score for each related query in the cluster is computed based on similarity and popularity. The method for computing the rank score is presented next in this section.
3. Finally, the related queries are returned ordered according to their rank score.

The rank score of a related query measures its interest and is obtained by combining the following notions:

***Similarity of the query*:** measuring the similarity of the query to the input query is depended on the relation between queries and URLs. Each query is considered as a vector, depicted in equation (1), which the features are relation between the query and URLs computed with equation (2), so we measure the similarity between two queries with cosine similarity.

Fig. 3, for example, displays how queries are considered as vectors. The cosine similarity measure is accomplished by equation (4), where the numerator represents the inner product of two query vectors; furthermore, the dominator is the product of their length.

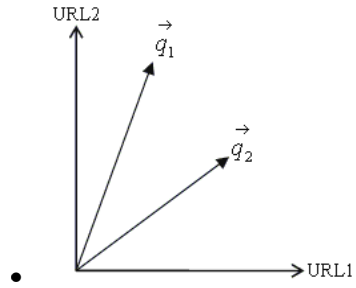


Fig. 3. Representing queries as vectors in a space that every dimension denotes a URL

$$Sim(q_i, q_j) = \frac{\overrightarrow{q_i}.\overrightarrow{q_j}}{|\overrightarrow{q_i}| \times |\overrightarrow{q_j}|} \quad (4)$$

Two different situations can be occurred for the input query. Once the input query has been also issued previously, so measuring similarity can be done based on related information in query log. In the other case, the input query is new, and so far it has not been appeared in query log. In this situation, to find relevant cluster, firstly based on notation (2), mentioned in related work, we will attempt to find a query with the highest similarity, so the cluster of such query can be considered as the relevant cluster.

***Support of the query**:* this is a measure of how the query is relevant in the cluster. We measure the support of the query as the fraction of the number of URLs returned by the query that captured the attention of users (clicked URLs). It is estimated from the query log and computed as equation (5), where $|L_i|$ is the number of clicked URLs for q_i that is member of cluster *C*.

$$Sup(q_i) = \frac{|L_i|}{\sum_{j \in C} |L_j|} \quad (5)$$

The similarity and support of a query can be normalized, and then linearly combined, yielding the rank score of the query. In our experiment, we use equation (6) to compute rank score for recommended queries for the input query (q).

$$Rank(q_i) = \alpha \times Sim(q_i, q) + \beta \times Sup(q_i) \quad (6)$$

4 Experimental Evaluation

A query log of the AOL search engine was used for collecting clickthrough data. This collection consists of ~20M web queries collected from ~650k users over three months from 1 march to 31 may 2006 [12]. In order to evaluating our similarity measure, we compared it with a similarity measure which has been used in Beeferman and Berger [3] (after that we refer it with Beeferman) which *Jaccard Similarity Coefficient* [9] has been used to measure similarity between queries. We extracted 10,000 queries from the AOL data set in order to accomplish the evaluation. After constructing clusters, ten queries of the clustered dataset were selected: (1) lottery; (2) weather; (3) Vietnam; (4) the child Wonderland Company; (5) budget truck rental; (6) holiday mansion houseboat; (7) CBC companies; (8) ford; (9) zip codes and (10) Georgia guardianship laws. All selected queries had been sent to the recommendation algorithm in order to suggest helpful queries. The equation (6) was used for ranking, where we set α to 0.8 and β to 0.2. To appear how recommendation is done, the ranking suggested to query 3 (World War 2) has been depicted in Table 1. The queries are ordered descendingly according to their relation with the input query computed by equation (5). The result show that the algorithm discovered semantically related queries that are built upon different keyword. As an example, for a non-expert user the

Table 1. Suggested Queries for query "World War 2"

Suggested Query	Similarity	Support	Rank Score
world war 2 facts	0.995	0.24	0.844
world war 2 weapons	0.982	0.18	0.8216
Hitler	0.801	0.29	0.6988
pearl harbor	0.841	0.1	0.6928
atomic bomb	0.732	0.29	0.6436
hydrogen bomb	0.721	0.14	0.6048
cold war	0.501	0.1	0.4208

keyword "*pearl harbor*" may be unfamiliar for investigating about the major factor that motivated the United States to join World War 2, or one may wants to familiar with weapons that are used in world war 2, so the query "*atomic bomb*" or *" hydrogen bomb"* can be useful. Notice that our algorithm found queries with related terms, some of which would be difficult to use for users.

5 Conclusion

We have described a method for recommending associated queries according to a clustering process over web queries extracted from a search engine query log. Currently we are performing experimentations with larger query logs and considering more queries, to improve the experimental assessment of our approach. Moreevere, we are attempting to do query expansion using the keywords related to clusters. We also consider improving the notion of similarity of a query. One direction for doing so is also bearing in mind the relationship between terms of queries and combining it with the current similarity measure. As future work, we will try to improve the notion of attention of the recommended queries and to expand other notions of interest for the query recommender system. For example, finding queries share words but not clicked URL's. This might involve that the same words have different meanings if the text of the URL's also is not shared. Hence we can detect polysemic words.

Acknowledgement

The research work done in this paper is financially supported by the Iranian Telecommunication Research Center (ITRC) under the research protocol, T/500/1895, on May 7 2007.

References

1. Yates, R.B.: Query usage mining in search engines. In: Scime, A. (ed.) Web Mining: Applications and Techniques. Idea Group (2004)
2. Yates, R.B., Ribeiro-Neto, B.: Modern Information Retrieval, ch. 3, pp. 75–79

3. Beeferman, D., Berger, A.: Agglomerative clustering of a search engine query log. In: KDD, Boston, MA USA, pp. 407–416 (2000)
4. Fonseca, B.M., Golgher, P.B., De Moura, E.S., Ziviani, N.: Using association rules to discovery search engines related queries. In: First Latin American Web Congress (LAWEB 2003), Santiago, Chile (November 2003)
5. Jansen, D., Spink, A., Bateman, J., Saracevic, T.: Real life information retrieval: a study of user queries on the web. ACM SIGIR Forum 32(1), 5–17 (1998)
6. Wen, J., Nie, J., Zhang, H.: Clustering user queries of a search engine. In: 10th International World Wide Web Conference. W3C, pp. 162–168 (2001)
7. Hosseini, M., Abolhassani, H.: Hierarchical Co-Clustering for Web Queries and Selected URLs. In: Benatallah, B., Casati, F., Georgakopoulos, D., Bartolini, C., Sadiq, W., Godart, C. (eds.) WISE 2007. LNCS, vol. 4831, pp. 653–662. Springer, Heidelberg (2007)
8. Zaiane, O.R., Strilets, A.: Finding similar queries to satisfy searches based on query traces. In: Proceedings of the International Workshop on Efficient Web-Based Information Systems (EWIS), Montpellier, France (September 2002)
9. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge (2007)
10. Golub, G., Van Loan, C.: Matrix Computation, Johns Hopkins, Baltimore, 2nd edn. (1989)
11. Zhao, Y., Karypis, G.: Evaluation of hierarchical clustering algorithms for document datasets. In: CIKM, pp. 515–524. ACM, New York (2002)
12. Pass, G., Chowdhury, A., Torgeson, C.: A Picture of Search. In: 1st International Conference on Scalable Information Systems, Hong Kong. AOL (June 2006)

Reliability Evaluation in Grid Environment

Saeed Parsa and Fereshteh Azadi Parand

Iran University of Science and Technology (IUST), PO Box 16846_13114,
Narmak, Tehran, Iran
{parsa,parand}@iust.ac.ir

Abstract. Service time especially for tasks with predefined deadline is an important criterion in evaluation of a grid resource manager. This parameter is affected directly by the failure of connection links and computational resources. In this paper a new approach to estimate expectation of execution time and probability of successful execution of a task considering both permanent and transient failures is proposed. Also in current works in order to increase the probability of successful execution and consequently to decrease the execution time, the use of parallel resources is suggested. Nevertheless due to the cost constraint the use of parallel resources is not recommended. In this paper the use of active parallel systems to increase the reliability of services in a grid environment is suggested. An active parallel system has lower impact on reducing the execution time, but is more economical.

Keywords: transient failure, reliability evaluation, computational grid, resource management.

1 Introduction

Grid computing [9] has emerged as an important new field, distinguished from conventional distributed computing by its focus on large-scale resource sharing, innovative applications, and, in some cases, high-performance orientation[10] [11]. Computational grid is built on Internet protocols and services to support the creation and use of computation environments.

Grid resource managers [12] [19],the heart of grid, is responsible for task division and submission to available grid resources. Service time, which is the index of the performance [1] of grid resource managers, is a random variable [8][21]. The execution time of a service task, service time, is a function of the execution time of its subtasks. The execution time of each sub-task depends on the availability and speed of the resources selected by the resource manager to perform the requested service. Obviously, the lower the reliability of the resources engaged to carry on a subtask, the lower the probability of getting an appropriate response in predefined deadlines. Therefore, accurate estimation of reliability has significant importance in grid performance analysis.

The study of reliability in grid environments due to the large scale distributed structure of grid networks, heterogeneously of resources, inexistence of central control on resources and connection links and in one word unreliable nature of grid is

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 388–395, 2008.

comparatively complicated. There are some research works on grid service reliability. In [5], a model to analyze the reliability of services within wide-area distributed systems, which are one of the ancestors of the grid systems, is presented. The function of control center in this model is similar to that of RMS in grid environments. Most traditional reliability models [13][3][4][18] are based upon this assumption that the operational and failure state probabilities of computational resources and their links are approximately constant. This assumption in grid environment, due to the massive data transmission and large volume of computations, can not be considered correct. Based upon the fact that hardware resources and connections failure number usually follow a poisson distribution, the reliability function R(t) and failure distribution f(t) are both considered exponential [15][6][7][14].

The major problem in [15][6][7][14][16] is the permanent failure assumption. In practice, it is observed that the failure of computational resources and connections are transient. For example, most often, though not always, after a number of unsuccessful attempts data may be transferred via the network.

In this paper, a new set of equations to analyze the reliability and time of grid services, considering transient failures, is proposed.

Current approaches for analysis of grid service reliability assume each service task to be assigned either to a separate resource or to multiple resources, simultaneously. In practice, due to the cost constraints, the use of passive parallel resources is more acceptable than the use of parallel resources. In this paper reliability of a service task considering passive parallel resources is performed.

The remaining parts of this paper are organized as follows: in Section 2-1, all the assumptions underlying our proposed reliability model are clarified. In Section 2-2, considering reliability issues the execution time of. sub-tasks of a task, assigned to RMS, is analyzed. Section 2-3 presents a discussion of the analysis of services reliability considering existence of passive parallel resources.

2 Proposed Model

Before presenting our proposed grid reliability analysis model, assumptions about the virtual networks, described above, including computational resources, connection links, and resource manager are given

2.1 Assumptions

1. When a resource or communication link is available for processing a subtask, it has constant processing speed.
2. Each resource is directly connected to the RMS by single communication channel.
3. Each subtask processing time is proportional to its computational complexity.
4. The failure rate of communication channels and resources in hot standby mode and active mode is assumed to be the same. Also, the failure type of processor is supposed to be fail-stop [20]. Fail-stop processor only halts, in other words nodes that detect internal faults simply stop working in a way detectable from the outside. The only visible effects of failure in a fail- stop processor are to stop execution and lose internal state and contents of the connected volatile storage.

Nomenclature	
RMS	Resource manager system
a_j	Amount of data transmitted between the RMS and the resource processing subtask j
C	Computational complexity of the entire task
c_j	Computational complexity subtask j
λ_k	Failure rate of resource k
π_k	Failure rate of communication channel k
θ	Random time of task execution by the system (service time)
θ_i	Computational time of task by resource ith
θ^*	Maximum allowed service time
$R(\theta^*)$	probability that service time is less than θ^*
x_k	processing speed of resource k
s_k	Data transmission speed of communication channel k
$E[Tsucc_{A_i}^{(j)}]$	Expectation of execution time of subtask A_i provided it is finished in j^{th} try
$Psucc_{A_i}^{(j)}$	Probability of successful execution of subtask A_i in j^{th} try
$Psucc_{A_i}^{\theta^*}$	Probability of successful execution of subtask A_i in predefined deadline.
$ProbF_L$	Probability of link failure/Probability of failure
$ProbF_N$	Probability of node failure/Probability of failure
$E[time\_of\_failure \mid failure\_occur^{(i)}]$	Expectation of failure time provided that the failure is occurred in i^{th} try

5. If resource failure occurs during subtask processing or channel failure occurs during data transmission between the resource and the RMS, the subtask will fail.
6. If resource failure occurs during subtask processing or channel failure occurs during data transmission between the resource and the RMS and before the task termination deadline, the task will be reassigned to the resource.
7. Failures at different resources and communication channels are independent.
8. Grid resource manager is fully reliable.
9. The time of task processing by the resource manager (division into subtasks, sending the subtasks to the resources, receiving the results and integrating them into entire task output) is negligible when compared with subtasks processing time.

2.2 Estimation of Sub-task Completion Time

Before the completion time of a task can be computed, the completion time of all of its subtasks has to be computed. The completion time of each of the sub-tasks is a function of its computational complexity, allocated resources (computation resource, link) speed, and reliability.

$$T_{succ}(sub\_task_i) = f(s_k, x_k, \lambda_k, \pi_k, c_i, a_i) \quad k = Index\ of\ \ kth\ used\ resource \quad (1)$$

If in the case of a failure the resource manager keeps retrying to assign the failed subtask to the failing resource before a predefined deadline or successful accomplishment of the subtask then the expected execution time of the subtask A_j, $E[Tsucc_{A_j}]$, can be calculated as follows:

$$E[Tsucc_{A_j}] = \sum_{i=1}^{N} Psucc_{A_j}^{(i)} * E[T_{succ}{}_{A_j}^{(i)}]$$

$$where \qquad \sum_{i=1}^{N} Psucc_{A_j}^{(i)} \le 1 \quad and \quad E[Tsucc_{A_j}^{(N)}] < \theta^* \quad (2)$$

In the above relations, N indicates the number of attempts to assign the subtask before a predefined deadline, θ^*, $Psucc_{A_j}^{(i)}$ is the probability of success in ith try and $E[T_{succ}{}_{A_j}^{(i)}]$ is the average execution time provided that there is a successful try after i-1 unsuccessful tries. In order to calculate $E[Tsucc_{A_k}]$ first we show how the value $Psucc_{A_j}^{(1)}$ and $E[T_{succ}{}_{A_j}^{(1)}]$ can be computed.

The successful operation of a physical resource with constant failure rate, λ, for a period of time, t, can be calculated as follows [20].

$$P_{succ} = e^{-\lambda t} \quad (3)$$

In addition, the first try can be successful if the resource to be in operation until to finish the assigned job and the connection to be in operation until to send the result completely. As a result probability of successful execution of subtask A_k on resource can be calculated as follows:

$$Psucc_{A_j}^{(1)} = e^{-\lambda c_j / x_k} * e^{-\pi a_j / s_k} \quad (4)$$

Where c_j / x_k is the processing time of sub-task j on resource k and a_{j/s_k} is the data transmission time of task j on channel k.

The time required to execute and receive the results of each subtask, provided that the assigned computational resource accomplishes the subtask successfully in the first try on resource kth, is calculated as follows:

$$E[T_{succ}{}_{A_j}^{(1)}] = c_j / x_k + a_j / s_k \quad (5)$$

Subject to: $(a_j / s_k + c_j / x_k) \le \theta^*$

Where θ^* is the subtask execution deadline.

In order to calculate the subtask execution time in a second try, first the proportion of resource and connection failures on the total failure is computed

$$\Pr obF_{N=} \lambda / (\lambda + \pi) \quad (6)$$

$$\Pr obF_L = \pi/(\lambda+\pi) \quad (7)$$

In addition, we know that the expectation time of the resource functioning conditionally that we know a failure occurred in first try will be obtained as follows:

$$E[time\_of\_failure_N \mid failure\_occurance_N^{(1)}] = \frac{\int_0^{c_j/x_k} (\lambda e^{-\lambda t}) t dt}{\int_0^{c_j/x_k} \lambda e^{-\lambda t} dt} \quad (8)$$

The above relation can be simplified as: $1/\lambda + \frac{e^{-\lambda c_j/x_k} * c_j}{(e^{-\lambda c_j/x_k} - 1) * x_k}$ (9)

If a link failure occurs in the first try, then the expectation time of the link functioning before its first failure will be obtained as follows:

$$E[time\_of\_failure_L \mid failure\_occurance_L^{(1)}] = \frac{\int_0^{a_j/s_k} \pi e^{-\pi t} t dt}{\int_0^{a_j/s_k} \pi e^{-\pi t} t dt} \quad (10)$$

which can be simplified as: $1/\pi + \frac{e^{-\pi a_j/s_k} * a_j}{(e^{-\pi a_j/s_k} - 1) * s_k}$ (11)

In general, the expectation time of failure for each subtask can be computed as follows:

$$E[time\_of\_failure \mid failure\_occurance^{(1)}] =$$
$$\Pr obF_N * E[time\_of\_failure_N \mid failure\_occurance_N^{(1)}] +$$
$$\Pr obF_L * E[time\_of\_failure_L \mid failure\_occurance_L^{(1)}] \quad (12)$$

Its assumed that in the case of failure the RMS keeps resubmitting its requests to the resource assigned to execute a given subtask until it succeeds or reaches a predefined deadline, $\theta*$. It is also assumed that the period of resubmission is equal to T_{succ_1}, which could be computed by applying relation (5), described above, With the above assumptions the value of P_{succ_2} and T_{succ_2} can be computed as follows:

$$Psucc_{A_j}^{(2)} = (1 - Psucc_{A_j}^{(1)}) * Psucc_{A_j}^{(1)} \quad (13)$$

$$E[T_{succ\,A_j}^{(2)}] = E[T_{succ\,A_j}^{(1)}] + \left[(E[time\_of\_failure \mid failure\_occurance^{(1)}] + MTTR) / E[T_{succ\,A_j}^{(1)}] \right]$$
$$* E[T_{succ\,A_j}^{(1)}] \qquad \text{Subject to: } E[T_{succ\,A_j}^{(2)}] < \theta^* \quad (14)$$

In general, the probability of a successful execution and transmission after m-1 unsuccessful tries can be calculated as follows:

$$Psucc_{A_j}^{(m)} = (1 - e^{-\pi a_j/s_k} e^{-\lambda c_j/x_k})^{m-1} * e^{-\lambda c_j/x_k} * e^{-\pi a_j/s_k} \quad (15)$$

The average response time, T_{succ_m}, takes the form

$$E[T_{succ_{A_j}}^{(m)}] = (m-1)*$$

$$\left\lceil (E[time\_of\_failure \mid failure\_occurance^{(1)}] + MTTR) / E[T_{succ_{A_j}}^{(1)}] \right\rceil *$$

$$E[T_{succ_{A_j}}^{(1)}] + E[T_{succ_{A_j}}^{(1)}] \qquad (16)$$

$$\text{Subject to: } E[T_{succ_{A_j}}^{(m)}]_{<\theta^*}$$

Therefore, the probability of successful execution in predefined deadline, $P_{succ_{\theta^*}}$, is obtained as follow:

$$Psucc_{A_j}^{*} = \sum_{m=0}^{m_u} (1 - e^{-\pi a_j / s_k} e^{-\lambda c_j / x_k})^m * e^{-\lambda c_j / x_k} * e^{-\pi a_j / s_k} \quad (17)$$

Which can be simplified as:

$$Psucc_{A_j}^{*} = 1 - (1 - e^{-\pi a_j / s_k} e^{-\lambda c_j / x_k})^{m_u + 1} \qquad (18)$$

$$E[T_{succ_{A_j}}^{(m_u)}] \le \theta^*$$

$$m_u \le (\theta^* - E[T_{succ_{A_j}}^{(1)}]) / (E[T_{succ_{A_j}}^{(2)}] - E[T_{succ_{A_j}}^{(1)}]) + 1 \quad (19)$$

Considering the number of failures before a subtask can be executed properly, the time required for a successful execution of the subtask is a random variable. The expectation time for a successful execution is computed as follows:`

$$T_{succ_{\theta^*}} = \sum_{m=0}^{m_u} Psucc_{A_j}^{(m)} * E[T_{succ_{A_j}}^{(m)}] (20)$$

$$m_u \le (\theta^* - E[T_{succ_{A_j}}^{(1)}]) / (E[T_{succ_{A_j}}^{(2)}] - E[T_{succ_{A_j}}^{(1)}]) + 1$$

2.3 Estimation of Task Completion Time

The execution time of a task depends on the configuration of its subtasks and the way in which resources are utilized. Resources may be utilized in three ways of:

(1) Assign each subtask to a separate resource with no alternate resources
(2) Assign a task or a fraction of its subtasks to more than one resource at a same time
(3) Assign each subtask to a separate resource with an alternate resource which may be used in parallel with the primary resource in the case of transient failure

If each subtask is assigned only to a single resource without any alternative resources then probability of successful execution and expectation of task execution time will be calculated as follows:

$$Psucc_A^{\theta^*} = \prod_{k=1}^{N} Psucc_{A_k}^{\theta^*}, A_k = kth\_subtask\_of\_task\ A \quad (21)$$

$$E[Tsucc_A^{*}] = \max(E[Tsucc_{A_k}^{*}]) \qquad (22)$$

Otherwise if each subtask is assigned to more than one resource concurrently then probability of successful execution and expectation of task execution time will be calculated as follows:

$$Psucc_A^{\theta^*} = 1 - \prod_{k=1}^{N} (1 - Psucc_{A_k}^{\theta^*}) \qquad (23)$$

$$E[Tsucc_A^{\theta^*}] = \min(E[Tsucc_{A_k}^{\theta^*}]) \qquad (24)$$

Sometimes because of cost constraints, the use of concurrent resources is not economical. In these situations it would be beneficial to apply alternative resources in the case of failures or in the other words apply passive parallel resources.

Suppose that there are m resources in passive parallel mode such that if the ith resource fails, then its assigned subtask will be delivered to the i+ resource which may run the subtask in parallel with the resource itself. This process of allocating alternate resources continues until there are no alternate resources nor a predefined deadline is reached. In this state the successful probability of a task execution and expectation time of service is calculates:

$$Psucc_A^{\theta^*} = \sum_{k=1}^{N} \left(\left[\prod_{i=1}^{k} (1 - Psucc_{A_i}^{\theta^*})^{k-i+1} \right] * \left[1 - \coprod_{l=1}^{k} (1 - Psucc_{A_i}^{\theta^*}) \right] \right) \qquad (25)$$

$$E[Tsucc_A^{\theta^*}] = \sum_{k=1}^{N} \max_{i=1}^{k} \{ \frac{c_i}{x_k} + \frac{a_i}{s_k} \} (26)$$

$$where \qquad T_{succ} \le \theta^*$$

3 Conclusion

The time it takes to perform a task assigned to a grid resource manager and the probability of the successful completion of the assigned task by a predefined deadline are two important parameters in performance analysis of grid. The values of these two parameters are affected by the reliability of the resources and their connections with the grid resource manager. The reliability itself is based upon the behavior of transient and permanent failure rate functions. Therefore, the mathematical models of reliability should consider both transient and permanent failures. In this paper, a new mathematical model for estimating execution times of subtasks is proposed. The estimated execution times of subtasks are applied according to their distribution model over the grid resources to estimate the execution time and the probability of successful execution of their enclosing task in predefined deadline.

References

1. Abramson, D., Giddy, J., Kotler, L.: High performance parametric modeling with Nimrod/G. In: 14th International parallel and distributed processing symposium, pp. 520–528 (2000)
2. Chang, M.S., Chen, D.J., Lin, M.S.: The distributed program reliability analysis on a star topology. Computers and Operations Research 27(2), 129–142 (2000)

3. Chen, D.J., Huang, T.H.: Reliability analysis of distributed systems based on a fast reliability algorithm. IEEE Trans. Parallel Distribut. Syst. 3(2), 139–154 (1992)
4. Chen, D.J., Chen, R.S., Huang, T.H.: A heuristic approach to generating file spanning trees for reliability analysis of distributed computing systems. Comput. Math. Appl. 34, 115–131 (1997)
5. Dai, Y.S., Xie, M., Poh, K.L., Liu, G.Q.: A study of service reliability and availability for distributed systems. Reliability Eng. Syst. Safety 79(1), 103–112 (2003)
6. Dai, Y.S., Levitin, G.: Reliability and Performance of Tree-structured Grid Services. IEEE Transactions on Reliability 55(2), 337–349 (2006)
7. Dai, Y.S., Levitin, G., Trivedi, K.S.: Performance and Reliability of Tree-Structured Grid Services Considering Data Dependence and Failure Correlation. IEEE Transactions on Computers (accepted, 2006)
8. England, D., Weissman, J.B.: A stochastic control model for the deployment of dynamic grid services. In: The 5th IEEE/ACM InternationaWorkshop on Grid Computing, pp. 192–199 (2004)
9. Foster, I., Kesselman, C.: The Grid 2: Blueprint for a New Computing Infrastructure. Morgan-Kaufmann, Los Alios (2003)
10. Foster, I., Kesselman, C.: The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann Publishers, Los Altos (1998)
11. Foster, I., Kesselman, C., Tuecke, S.: The anatomy of the grid: enabling scalable virtual organizations. International J. Supercomputer Applications 15(3) (2001)
12. Krauter, K., Buyya, R., Maheswaran, M.: A taxonomy and survey of grid resource management systems for distributed computing. SoftwarePractice and Experience 32(2), 135–164 (2002)
13. Kumar, R., Hariri, S., Raghavendra, C.S.: Distributed program reliability analysis. IEEE Trans. Software Eng. SE-12, 42–50 (1986)
14. Kumar, A.: Adaptive load control of the central processor in a distributed system with a star topology. IEEE Transactions on Computers 38(11), 1502–1512 (1989)
15. Levitin, G., Dai, Y.S.: Service reliability and performance in grid system with star topology. Reliability Engineering and System Safety 92(1), 40–46 (2007)
16. Levitin, G., Dai, Y.S.: Service reliability and performance in grid system with star topology. Reliability Engineering and System Safety 92(1), 40–46 (2007)
17. Lewise, E.E.: Intoduction To reliability Engineering. John Wiley & Son, Chichester (1994)
18. Lin, M.S., Chang, M.S., Chen, D.J., Ku, K.L.: The distributed program reliability analysis on ring-type topologies. Comput Oper. Res. 28, 625–635 (2001)
19. Nabrzyski, J., Schopf, J.M., Weglarz, J.: Grid resource management. Kluwer Publishing, Dordrecht (2003)
20. Schlichting, R.D., Schneider, F.B.: Fail-Stop Processors: An Approach to Designing Fault-Tolerant Computing Systems. ACM Trans. Comput. Syst. 1(3), 222–238 (1983)
21. Shan, H., Oliker, L., Biswas, R.: Job superscheduler architecture and performance in computational grid environments. In: Proceedings of the ACM/IEEE Supercomputing 2003 Conference (SC 2003), 2003, pp. 44–59 (2003)

Formulating Priority Coefficients for Information Quality Criteria on the Blog

Mohammad Javad Kargar[1], Abd R. Ramli[2], H. Ibrahim[3], and F. Azimzadeh[4]

[1] Department of Computer Engineering, Islamic Azad University of Maybod, Maybod, Iran
showkaran@hotmail.com
[2,4] Department of Computer Engineering, University Putra Malaysia, 43400 Serdang, Malaysia
arr@eng.upm.edu.my, f.azimzadeh@yahoo.com
[3] Faculty of Information Technology and Computer Science, University Putra Malaysia, 43400 Serdang, Malaysia
hamidah@fsktm.upm.edu.my

Abstract. The World Wide Web (WWW) has become one of the fastest growing electronic information sources. Meanwhile new facilities for producing web pages such as Blogs make this issue more significant because Blogs have simple content management tools enabling non-experts to build easily updatable web diaries or online journals. On the other hand despite a decade of active research, information quality lacks comprehensive methodology for its assessment and improvement. Especially there is not any framework for measuring information quality on the Blogs yet. This paper establishes a survey on Iranian Blog as our case study presents results of the survey, prioritizes information quality criteria by allocating priority coefficient to all the criteria as an important prerequisite for measuring information quality in the Blogs. Also results of the research by gap analysis shows there is a solid consensus between Bloggers and visitors about the priorities of information quality criteria in the Blogs.

Keywords: Blog quality, web quality, information quality, gap analysis.

1 Introduction

The vast amount of information on the World Wide Web is created and published by many different types of providers, including businesses, organizations, governments, and individuals. Unlike books and journals, most of this information is unfiltered, i.e. not subject to editing or peer review by experts. This lack of quality control and the explosion of web sites make the task of finding quality information on the web especially critical.

In May 2007, Blog search engine Technorati tracking more than 70 million Blogs. Every day 120,000 new Blogs are created and 1.5 million posts are made, it found during its quarterly survey [1]. A Blog, sometimes written as web log or Weblog, is a Web site that consists of a series of entries arranged in reverse chronological order, often updated on frequently with new information about particular topics. The information can be written by the site owner, gleaned from other Web sites or other sources, or contributed by users.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 396–403, 2008.

Despite a decade of research and practice, only piece meal, ad hoc techniques are available for measuring, analyzing and improving Information Quality (IQ) on the web. Unfortunately there is not any framework for measuring IQ in Blogs. We believed that Blog can be a suitable application for evaluating quality of information because Blogs use common templates, so that quality of content of a Blog is almost equal to quality of Blog.

In this research we focus on Iranian Blogs. Farsi (or Persian), the official language of Iran, is the newcomer to the top 10 Blogging languages [1]. Although there is not exact statistic for number of Iranian Blogs, according to information which we collect from popular Blog server in Iran, there are around 2 million Blog in Iran. Interesting information collected by Alexa, a web information company, show that after Yahoo and Google, Blogfa as a Blog service provider is third and PersianBlog is seventh top sites in Iran. Blogging in Iran has grown so fast because it meets the needs no longer met by the print media; it provides a safe space in which people may write freely on a wide variety of topics.

The paper presents results from survey on Iranian Blog as our survey. Then are prioritized IQ criteria and is allocated priority coefficient for each of the criteria. The remainder of this paper is organized as follows. Section 2 presents an overview of the relevant literature. Section 3 describes methodology of the research. Next, in Section 4, is analyzed the results of the survey. Finally, Section 5 presents our conclusions.

2 Related Works

In our earlier research [2], we classified IQ researches into four categories; first, literatures which only have listed some of IQ criteria. For instance Collins Memorial Library[3] and Virtual Case [4] have listed some criteria. Second, researches which propose information quality models. These models are general purpose or special purpose. In general purpose model criteria are examined in a most general way. In the other word criteria selection and definition is independent of environment and information framework. The aim of such models is that everybody can match the model to their applications. TDQM [5], Naumann [6] and AIMQ [7] are most popular general purpose models.

Unlike general purpose models special purpose models develop the criteria according to their requirements in a specific application such as Data Warehouse Quality (DWQ) [8], IQIP for information retrieval purposes [9] and intranet application [10], quality of information in Wikipedia [11, 12].The aim of such models mainly hasn't been identifying criteria for information quality. Instead the models have been employed for efficiency improvement in considered application. Third, researches which have tackled a few of criteria and have attempted to find methods for computing and measuring the criteria. Measuring timeliness in [13, 14], cohesiveness in [15, 16], frequency analysis in [17, 18] are examples of these researches.

Forth, studies which propose frameworks for evaluating the quality of conceptual models. The aim of these researches is to identify worth and validity of information quality models. For instance, in [19] is conducted an empirical analysis of the conceptual model quality framework proposed by Lindland et al [20]. Although literature in information quality proposes several different techniques for measuring information

quality, none have addressed the issue of measuring and evaluating information quality in Blogs. There are researches such as [21] which have analyzed Blogs and [22, 23] which have studied Blog comments, without entering to information quality issue.

3 Research Design and Methodology

The main objective of this research is to prioritize and formulate information quality criteria in Blog. Before prioritization operation we need to prove our hypothesis for identified information quality criteria. In fact, information quality needs to be assessed within the context of its generation [24] and intended use [25]. This is because the attributes of data quality can vary depending on the context in which the data is to be used [26].

Performing surveys is a well-known strategy for doing empirical studies. The surveys have been used in computer sciences researches which are impacted by human perception such as software engineering [27, 28]. As well, quality is a matter of perception, and is often difficult to measure objectively. Like all other quality measures, it should be judged by the receiver. Following the design phase, a questionnaire for this Web-Based Survey was built using the HyperText Markup Language (HTML). After our Web-Based Survey was tested, its Web address was submitted to comment section of selected Blogs. When Blogers were clicking the link, were led to our special page. We briefly introduced our work in this page. 1500 person visited this page of 3000 invitation. After introducing the work in this page, visitors were led to questionnaire page. For more clarity we used Persian language in both pages. Of the 1500 visitors 790 visited the questionnaire page and 420 respondents answered the questions. According to [29, 30] this sample size is appropriate with confidence of 95%.

The questionnaire contained two major sections. The first section contained only one question which identified a respondent is a Blogger or not (we would liked to bring other questions but preferred to abandon them to keep respondents!). The second part of the questionnaire consisted of eighteen questions about information quality attitudes. We applied a standard procedure for the measurement of attitudes with a five grade Likert scale. Using Likert scale is widely practiced for measuring attitudes. Many of researches in various fields of information technology have used Likert scale among them recent researches in [31, 32]. For our 18-item questionnaire, each item was analyzed on a five-point Likert scale so that higher item scores indicated a more favorable attitude as 1 shows lowest priority and 5 highest priority. Of the 420 respondents, 230 were Bloggers while 70 were only Blog visitors. This numbers in turn show sizeable portion of Blog visitors are Bloggers themselves (here 75%). Cronbach's Alpha is used as an internal consistency technique to assess the homogeneity of the concepts in each category of the proposed research framework. Cronbach's Alpha is fairly standard in most discussions of reliability. In addition, it has been used successfully in other IS instrument development [25, 33]. A Cronbach's Alpha of 0.7 is thought to demonstrate good reliability although some authors report values as low as 0.5 to be satisfactory [34, 35]. Since Cronbach's Alpha for our questionnaire is 0.765, we conclude that the questionnaire developed in this study has high reliability.

4 Data Analysis and Prioritization

After demonstrating hypothesizes for information quality criteria by questionnaire, in this stage is prioritized the criteria by means comparison of the criteria according to respondents' points. Table 1 shows mean and standard deviation for each criterion.

Regarding that we are going to prioritize information quality criteria, we have to allocate a coefficient for each criterion according to obtained mean scores in Table 1. For this aim we mapped every mean to coefficient so that sum of all coefficients will be 1. Thus this method allocates to the each of criterion a priority coefficient. The highest score is for understandability and informativeness with mean of 4.42 and lowest score is for redundancy with mean 3.03. Standard deviation values show that most deviation is related to redundancy and customer support that both be placed in the low part of table. This deviation is normal because customer support and redundancy are completely subjective. Especially redundancy is controversial issue in Blog. Arising multimedia facilities on the web causes many of Bloggers use multimedia elements in their Blogs. This issue is more controversial where internet speed is usually low same as Iran.

Fig. 1 shows priority coefficient for information quality criteria. For instance highest coefficient is related to understandability with value .064 and lowest coefficient belong to redundancy with .043. In this case understandability has 67% more priority in compare of redundancy. However differences between some of criteria coefficients are not significant but between top and down coefficients this difference is significant.

Table 1. Mean and standard deviation for Information Quality criteria

Criteria	Mean	Std. Deviation	Criteria	Mean	Std. Deviation
Understandability	4.42	.693	Maintainability	3.84	1.031
Informativeness	4.42	.742	Availability	3.84	.926
Representation	4.17	.820	Authority	3.78	.936
Accuracy	4.14	.795	Latency	3.76	1.029
Completeness	4.05	.863	Popularity	3.64	.996
Timeliness	4.05	.919	Customer support	3.54	1.108
Believability	4.04	.852	Amount of Data	3.51	.918
Concise	4.00	.913	Objectivity	3.40	1.074
Cohesiveness	3.90	.897	Redundancy	3.03	1.111

4.1 Gap Analysis

IQ Role Gaps in general, compare the IQ assessments from respondents in different organizational roles, IS professionals, and information consumers. IQ Role Gaps is a useful diagnostic technique for determining whether differences between roles in a operational environment. The IQ assessment and comparison across roles serves to identify IQ problems and lays the foundation for IQ improvement [7].

Fig. 2 is an example of the Role Gap for the IQ using the data from our survey. There are two roles in Blog: Bloggers and visitors. The x-axis is the 18 criterions. The y-axis is the mean level of priority, which can range from1 to 5. The points in the

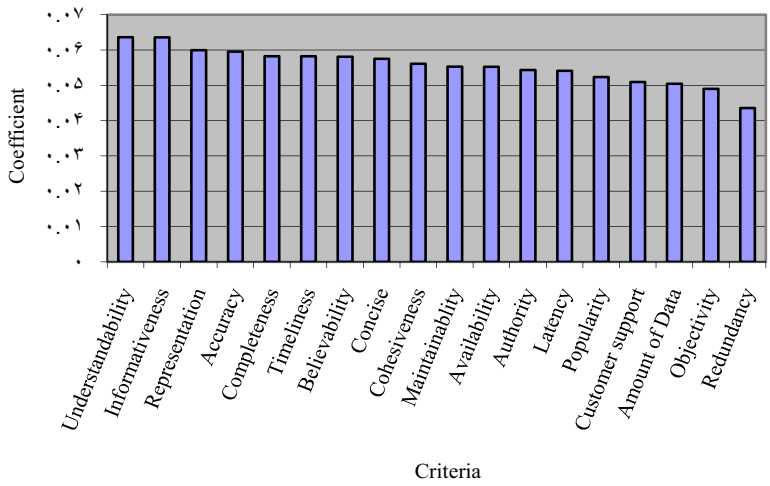


Fig. 1. Priority coefficients for Information Quality criteria

graph are the mean level of priority reported by visitors as information consumers (triangles) and the mean level reported by Bloggers as information producers (squares).When analyzing IQ Role Gaps, three indicators should be considered: Size of the gap area; Location of the gap and Direction of the gap (positive versus negative);

The least of the gap is related to popularity and believability which gap is less than .01. Most of the gap size is related to timeliness and amount of data around .34. However in this case when .34 divided by 5, overall gap is not significant(less than .07).The location of the gap for all the criteria is upper half the scales, which is quite good; whereas the location of the gap for redundancy criterion as lowest mean, which is also upper half is around 3. The location of gap for believability and informativeness is highest around 4.42. The direction of the gap is defined to be positive when IS professionals (here Bloggers) assess the level of IQ to be higher than information consumers (here visitors). Thus, maintainability, amount of data and latency have some positive gap. Timeliness has a negative gap.

A large positive gap means that IS professionals are not aware of problems that information consumers are experiencing. In general, contexts with a large positive gap should focus on reducing the problem by gaining consensus between IS professionals and information consumers. However in Blog we expected gap between information producers and consumers be insignificant because Bloggers as information producer usually are not a professional. Even may sometimes visitors be more professional than Bloggers. As a result Role Gap analysis on our survey confirms that there is a consensus between Bloggers and visitors for priority level of IQ criteria. This consensus facilitates formulating IQ assessment for Blogs.

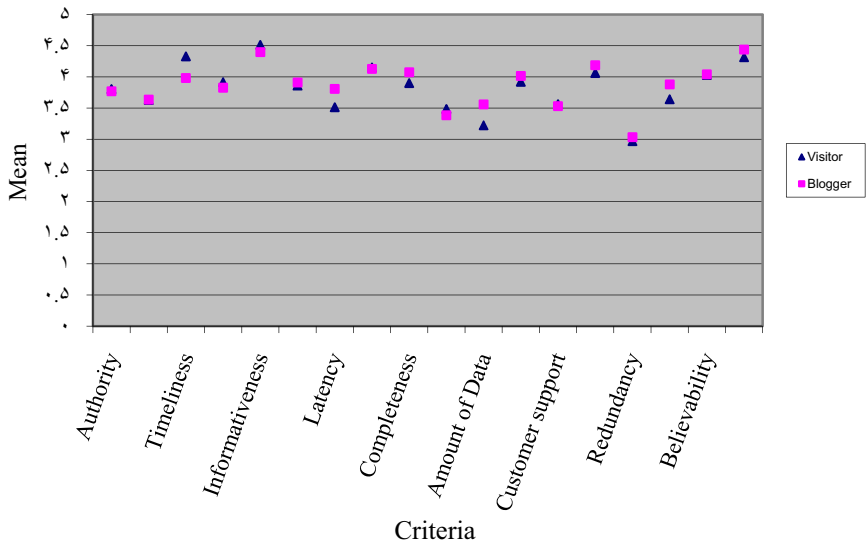


Fig. 2. Gap between visitors and Bloggers

Beside of gap analysis we tested criteria means for Bloggers and visitors. A test of normality showed all the means and sum of means are normal. For testing means between Bloggers and vistors used independent samples T- test because of normality of means and sum of means. Because significant value of equality of variance was .052 and this value is greater than .05, we considered equal variance. In this case value of significant (2-tailed) was .103 that is greater than .05. This value shows there is not significant difference for IQ priorities between Bloggers and visitors.

5 Conclusion

In this paper we established a survey on Iranian Blog in order to acquiring IQ criteria priorities. In this direction was calculated priority coefficient for each criterion. The analysis showed that however all the criteria have upper half scores but priority of some of the criteria are significant in compare with other criteria. The differences of priorities influence final quality of information score. This initial analysis of the priority of IQ quality of Blogs helps us to understand the ways in which quality is established and improved despite what seems at first glance the seemingly anarchic operation of the project. Also gained priority coefficients of IQ criteria can be employed for measuring quality of information on Blog in the next step of the project. As gap analysis shows there are a strong consensus between roles i.e. Bloggers and visitors about the priorities. This convergence makes easier IQ evaluation on Blogs.

In future research we plan to continue analyzing the quality of information in the Blogs both qualitatively and quantitatively to gain a better understanding of how IQ choices and assessments are made by the Blog community.

References

1. Sifry, D.: The State of the Live Web (2007)
2. Kargar, M.J., Ramli, A.A., Ibrahim, H., Noor, S.B.: Assessing Quality of Information on the Web Towards a Comprehensive Framework. In: 14th IEEE International conference on Internet Communication Technology (ICT/MICC), Malaysia, May 2007. IEEE, Los Alamitos (2007)
3. Ricigliano, L.: Criteria for Evaluating Information on the Web, vol. 2007. Collins Memorial Library (2006)
4. Tyburski, G.: Criteria for Quality in Information, vol. 2007 (2006)
5. Wang, R.Y., Strong, D.M.: Beyond accuracy: what data quality means to data consumers. Journal of Management Information Systems 12, 5–34 (1996)
6. Naumann, F., Rolker, C.: Assessment methods for information quality criteria. In: Proceedings of 5th International Conference on Information Quality, pp. 148–162 (2000)
7. Lee, Y.W., Strong, D.M., Kahn, B.K., Wang, R.Y.: AIMQ: a methodology for information quality assessment. Information & Management 40, 133–146 (2002)
8. Jarke, M., Vassiliou, Y.: Data warehouse quality design: A review of the DWQ project. In: Proceeding of the International Conference on Information Quality, Cambridge, MA (1997)
9. Knight, S.a., Burn, J.: Developing a Framework for Assessing Information Quality on the World Wide Web. Informing Science Journal 8, 159–172 (2005)
10. Leung, H.K.N.: Quality metrics for intranet applications. Information & Management 38, 137–152 (2001)
11. Stvilia, B., Twidale, M.B., Gasser, L., Smith, L.C.: Smith Information quality discussions in Wikipedia. In: Proceeding of the International Conference on Knowledge Management (ICKM 2005), pp. 1–20 (2005)
12. Stvilia, B., Twidale, M.B., Smith, L.C., Gasser, L.: Assessing information quality of a community-based encyclopedia. In: Proceedings of the International Conference on Information Quality (ICIQ), Cambridge, MA, pp. 442–454 (2005)
13. Zhang, Y., Zhu, H., Greenwood, S.: Empirical Validation of Website Timeliness Measures. In: Proceedings of the 29th Annual International Computer Software and Applications Conference (COMPSAC 2005), vol. 1, pp. 313–318. IEEE, Los Alamitos (2005)
14. Zhang, Y., Zhu, H., Huo, Q., Greenwood, S.: Measurement of Timeliness of Web-based Information Systems. In: Proceedings of the 6th World Multi-Conference on Systemic, Cybernetics and Informatics (SCI 2002) (2002)
15. Zhu, X., Gauch, S.: Incorporating quality metrics in centralized/distributed information retrieval on the World Wide Web. In: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, pp. 288–295. ACM, Athens (2000)
16. Zhu, X., Gauch, S., Gerhard, L., Kral, N., Pretschner, A.: Ontology-based web site mapping for information exploration. In: Proceedings of the eighth international conference on Information and knowledge management, pp. 188–194. ACM Press, New York (1999)
17. Dhyani, D., Ng, W.K., Bhowmick, S.S.: A Survey of Web Metrics. ACM Computing Surveys (CSUR) 34, 469–503 (2002)
18. Recker, M., Pitkow, J.: Predicting document access in large multimedia repositories. ACM Transactions on Computer-Human Interaction (TOCHI) 3, 352–375 (1996)
19. Moody, D.L., Sindre, G., Brasethvik, T., Solvberg, A.: Evaluating the Quality of Information Models: Empirical Testing of a Conceptual Model Quality Framework. In: 25th IEEE International Conference on Software Engineering (ICSE 2003), Portland, Oregon (2003)

20. Lindland, O.I., Sindre, G., Solvberg, A.: Understanding Quality in Conceptual Modeling. IEEE Software 3 (1994)
21. Herring, S.C., Scheidt, L.A., Bonus, S., Wright, E.: Bridging the Gap: A Genre Analysis of Weblogs. In: Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS 2004), vol. 4, pp. 40101–40102. IEEE Computer Society, Los Alamitos (2004)
22. Kumar, R., Novak, J., Raghavan, P., Tomkins, A.: On the bursty evolution of blogspace. In: Proceedings of the 12th international conference on World Wide Web, pp. 568–576. ACM Press, New York (2003)
23. Mishne, G., Glance, N.: Leave a reply: An analysis of weblog comments. In: Third annual workshop on the Weblogging ecosystem, Edinburgh, Scotland (2006)
24. Shanks, G., Corbitt, B.: Understanding data quality: Social and cultural aspects. In: Proceedings of the 10th Australasian Conference on Information Systems (1999)
25. Katerattanakul, P., Siau, K.: Measuring information quality of web sites: Development of an instrument. In: Proceedings of the 20th international conference on Information Systems, Charlotte, North Carolina, United States, pp. 279–285 (1999)
26. Shankar, G., Watts, S.: A relevant, believable approach for data quality assessment. In: Proceedings of 8th International Conference on Information Quality, pp. 178–189 (2003)
27. Punter, T., Ciolkowski, M., Freimut, B., John, I.: Conducting on-line surveys in software engineering. In: Proceedings Int. Symp. on Empirical Software Eng. 2003, pp. 80–88. IEEE Computer Society, Los Alamitos (2003)
28. Seaman, C.: Qualitative Methods in Empirical Studies of Software Engineering. Transactions on Software Engineering, 557–572 (1999)
29. Bartlett, J.E., Kotrlik, J.W., Higgins, C.C.: Organizational research: Determining appropriate sample size in survey research Information Technology, Learning, and Performance Journal 19 (2001)
30. Krejcie, R.V., Morgan, D.W.: Determining Sample Size for Research Activities. Educational and Psychological Measurement 30, 607–610 (1970)
31. Jennex, M.E., Smolnik, S., Croasdell, D.: Towards Defining Knowledge Management Success. In: Proceedings of the 40th Hawaii International Conference on System Sciences. IEEE Computer Society, Los Alamitos (2007)
32. Sharma, S., Singh, D., Agrawal, D.P.: Trust In Electronic Markets- Customer's Perspective 8 (2007)
33. Moore, G.C., Benbasat, I.: Development of an Instrument to Measure the Perceptions of Adopting an IT Innovation. Information Systems Research 2, 192–222 (1991)
34. Laukkanean, E., Halonen, P., Viinamaki, H.: Stability and internal consistency of the offer self-image questionnaire: A study of Finnish students. Journal of Youth and Adolescence 28, 71–77 (1999)
35. Staehr, L., Martin, M., Byrne, G.: Computer Attitudes and Computing Career Perceptions of First Year Computing Students. In: Proceedings of Informing Science 2001 – Bridging Diverse Disciplines, Krakow, Poland, pp. 502–509 (2001)

A Distributed Proxy System for High Speed Clients

Martin Krohn, Helena Unger, and Djamshid Tavangarian

University of Rostock
Faculty Computer Science and Electrical Engineering
Institute of Computer Science
Chair of Computer Architecture
Albert-Einstein-Strasse 21, 18059 Rostock, Germany
{firstname.lastname}@uni-rostock.de

Abstract. The use of Internet services with information access at anytime and anywhere, especially, for mobile clients at high speed, like cars or trains, has been increasing during the last years. This paper presents a novel multilayered architecture based on WiFi, WiMAX and proxy nodes offering reliability for broadband channels and Internet access on highways. We identify several weaknesses of standard fixed infrastructure-based networks and present novel extensions in order to improve several parameters for Quality of Service. Specifically, a distributed proxy system, algorithms, and a routing protocol supporting a feed-forward mechanism for efficient connection transfer between cells have been developed. Additionally, for further discussion of the bandwidth seam of WiFi and WiMAX, a simulation model with the OPNET Modeler was created and evaluated.

Keywords: vehicle-to-infrastructure (vehicle-to-roadside) communications, wireless technologies.

1 Motivation

The challenges of modern life and high mobility are a very intensive motivation for the development of novel cost efficient architecture based on Wireless Wide Area Networks (WWAN) and Wireless Local Area Networks (WLAN [1,2]) using roadside infrastructure supporting a seamless service provision within the highway environment.

2 Problem Definition

To support vehicle-to-infrastructure (vehicle-to-roadside) communications offering sufficient quality of service (QoS) for very aggressive mobility scenarios an optimized (possible cost efficient) network structure should be created, developed, realized, and evaluated. That system should provide a broad spectrum of seamless services to a large number of clients moving with very high speeds (up to 200 km/h) along a highway. Under broad spectrum of application we understand applications with different critical constrains from real-time up to broadband applications including the Triple-Play-Set.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 404–411, 2008.

3 System Model

In the following section we give a description of a generic abstract model for our network structure which is able to provide seamless services as described in the problem definition. The infrastructure for the network (refer to figure 1) consists of two layers: The backbone network and the access network. The backbone network has the task to provide the Internet access for the access network and is directly connected to an Internet Service Provider (ISP). The access network uses point-to-multi-point connections. For reasons of installation costs the supply network uses a wireless communication technology like WiMAX [2] (Worldwide Interoperability for Microwave Access). The backbone network consists of multiple wireless broadband receivers and senders. In WiMAX terms the receivers are called subscriber stations (SS). The senders are called base stations (BS), where multiple subscriber stations can connect to one base station (point-to-multi-point). The vehicle-access-infrastructure is implemented by using so called Points of Access (PoA). The PoAs are equipped with a WLAN interface which is able to accept connection requests and establish a seamless service providing connectivity to moving vehicles. The WLAN interface of the PoA establishes a so called WLAN cell characterized by a radius with a relevant strength of radio coverage according to the IEEE 802.11 standards [1,2]. For our consideration the area where the maximum bandwidth is provided is in the focus of interest. In our different versions of system models we use the ad-hoc as well as the infrastructure mode defined by the WiFi standard. The vehicles (clients) are also equipped with a WLAN interface. However, the PoA are more than only a WLAN interface, because it also may include additional intelligence supporting routing functionality, memory management, system load balancing, etc.

With the help of a WLAN/WiMAX backbone a communication cluster will be built around the so called cluster heads.

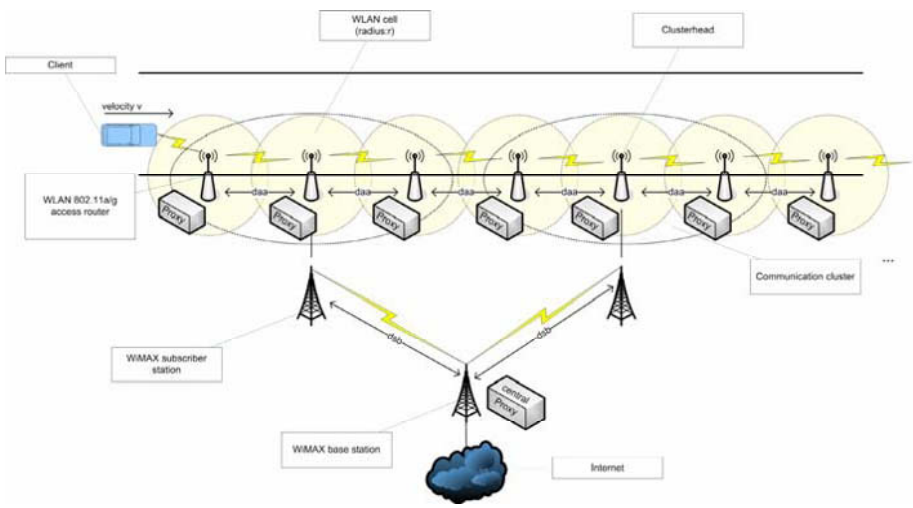


Fig. 1. Our network architecture with multilayered distributed proxy system

Where k>=0 is the dimension of a cluster. In general, k is the number of PoAs between two cluster heads. The PoA with the connection to the WWAN network is defined as a cluster head. Each PoA of the cluster is connected with its direct neighbours through a bidirectional WLAN connection. As an important feature, the network is self-organizing. By equipping a PoA with a WWAN interface the role of a clusterhead is assigned. Communication clusters are built up automatically by the PoAs, starting at the cluster heads.

The variation of k produces different network system models where one from each other differs in the arrangement of the supply as well as the access layer and implicitly in the properties. Independent from k the following criteria for a cost efficient and high level QoS supporting system should be taken in to account:

1. Hardware costs per kilometer
2. Infrastructure implementation costs (standard or extended with additional protocol and proxies
3. Quality of Service
   (a) Bandwidth
   (b) Latency, response delay
   (c) Packet loss
   (d) Hand off latency
   (e) Data throughput
   (f) Supported vehicle velocity for continuous network connection

With a growing of k the hardware costs per kilometer can be reduced. In this case the costs for the broadband receivers are saved for the cluster members not being cluster head. The number of cluster members is limited because, of a growing volume of data transfers within a cluster. The seams of two technologies, WLAN and WWAN providing different communication bandwidths, generate the problem of a bottleneck. To support an efficient use of the additional bandwidth when k is decreased and to relieve possible bottleneck problems at the cluster heads when k is increased, some additional effort for the model has to be taken into account. This variation of k is especially useful when the system shall be implemented in regions where a broadband ISP connection is only available very irregularly near the highway. The necessary extensions of the model are presented in the following section.

4 Network Architecture

The bandwidth WWAN technology is lower in comparison to the bandwidth of WLAN technologies. It is very likely that this situation will not change in the future. Concerning our system model, this means that there is a bottleneck within the backbone network. As a possible compensation, k can be reduced to 1, i.e. every PoA is directly connected to the WWAN.

In our system model we use current technologies. The WiMAX systems of today are able to transmit ca 4 MBit/s of downstream bandwidth [4]. Even with higher rates (e.g. 10 MBit/s downstream) the WWAN network bandwidth is inferior to the bandwidths offered by current WLAN standard technology like IEEE 802.11g with ca 24 MBit/s. As mentioned before, an increase of k introduces an additional bottleneck at the cluster heads within the access network.

The central connection point to the ISP is in contrast to the backbone network connection no bottleneck, as the geographical position can be chosen relatively flexible to gain a high bandwidth access to the ISP backbone.

To compensate the bottleneck within the WWAN and the WLAN we introduce a distributed data management control. The realization of such a system should be based on principals of pervasive communications which "will place side by side to traditional layered TCP/IP protocol suite, new cross-layering architecture, or architecture that completely eliminate layering and exploit application driven or data harnessing message forwarding, instead of delegating this responsibility to the network layer." [5].

The communicating system will use real best-effort and context aware message forwarding between system nodes (clients, PoAs, SSs, BSs). Basic context awareness can be realized if the system distinguishes between real-time and non real-time applications. More complex mechanism of context awareness should be realized due a more detailed analysis of application specific demands (see section 4.1). The high dynamic scenario dictates the modification of fixed end-to-end connectivity paradigm. The target system should master the following challenges:

- self-organization
- super dynamic routing
- fault-tolerance due to miss-estimation
- load-balancing
- spatial awareness and data harnessing inside each network node

According to listed above we introduce a distributed multilayered proxy system and modified communication protocols. The distributed multilayered proxy system consists of several proxies which are implemented at the central ISP access, within the PoAs of the cluster heads and within the remaining PoAs. The main idea of the introduction of proxies is to pre-load specific content packages (CP) which are requested by a client. A CP is one data subset which can be delivered to the client by exactly one PoA. The volume of a CP can be varied according to the user demands and network constraints. By avoiding each of the described bottlenecks, the client is able to download a content portion directly from the currently passed PoA. This so called feed-forward mechanism is described in detail in section 4.2.

By using a specialized routing mechanism the network load can be distributed in a way that the network is loaded maximally. The super dynamic routing algorithm has to support the novel feature of dynamic endpoint localization in order to support moving clients. It has to use a load-dependent metric for the load-balancing combined with fault-tolerance due to possible errors of the client position estimation.

According to the problem definition the system shall support real-time interactive services. In conclusion, the pre-loading and caching by proxies is not suitable. Thus, the proxies have to handle real-time traffic in a different way. A detailed discussion is presented in the following subsection.

4.1 Application Specific Demands

Applications have different demands on deadlines for the packet delivery. Voice over IP applications with a very good quality have for example a soft real-time deadline of

150 ms [6]. Other applications like shout-cast radio possibly have longer deadlines (i.e. up to 1 min). In contrast to real-time applications there are applications which have a demand on bandwidth and the time of arrival is a relatively unimportant factor. This kind of traffic (also known as best-effort traffic) can be modeled by using a very long deadline. There are two factors influencing the amount of pre-loading by the proxy system: The consumed bandwidth and the packet delivery deadlines. Thus, our network architecture must support different deadlines for different kinds of traffic. In dependence on the recognized traffic the proxies have to use different strategies. The strategy spectrum starts at non-caching for real-time traffic and ends at maximum caching for broadband application with long deadlines.

As an example, VoIP traffic cannot be cached because of its short deadline. In this case the traffic has to pass the proxy system unmodified. The mentioned bottlenecks of the network architecture are irrelevant to the low bandwidth VoIP traffic. Thus, there is no advantage of pre-loading VoIP traffic.

In difference to this, applications like video on demand with its high bandwidth demands can't be realized with the network architecture without a caching mechanism. The next subsection describes the caching within our architecture.

4.2 Multi Layer Distributed Proxy Server

As discussed in the last section several service applications need caching to be provided within our network architecture. As discussed above, we propose a multilayered distributed proxy system for this task. To achieve a good scalability we distribute data management control. The first layer proxy is called the central proxy. It is responsible for the download of the content, requested by any client and forms the CPs according to its bandwidth calculations. The second layer consists of cluster head proxies. They receive and forward CPs to the dynamic endpoints. The endpoint should be estimated using only local information contained by the CP.

The feed-forward mechanism (refer to subsection 4.3) routes the CPs to the clients. Each PoA caches a client associated CP which can be accessed by the addressed client.

4.3 A Feed-Forward Mechanism

An important extension for the presented architecture is a feed-forward mechanism. The main idea is to predict the future vehicle position using a determined vehicle velocity. The velocity is a key parameter for the estimation of the time a client spends within the WLAN cell of each PoA. This leads to a calculation of the volume of data consumed at the PoAs. The requested data can be partitioned according to the collected information. In figure 2 a sequence diagram of the feed-forward mechanism is given. At first, the client which is identified by its MAC address requests content. The responsible PoA records client associated data and does a calculation of the velocity. Together with the client related record the request is passed to the proxy server of the next layer. On receiving the request the proxy will establish an Internet connection and starts the download of the content. The proxy is able to examine the content's bit rate for streaming data and the total value of the content size. The proxy is now able to calculate a valid data distribution to support seamless services. Afterwards the proxy schedules uploads of CPs to the local proxies within the PoAs. A scheduling is important due to the bottleneck of the WWAN connection from the central proxy to the PoAs.

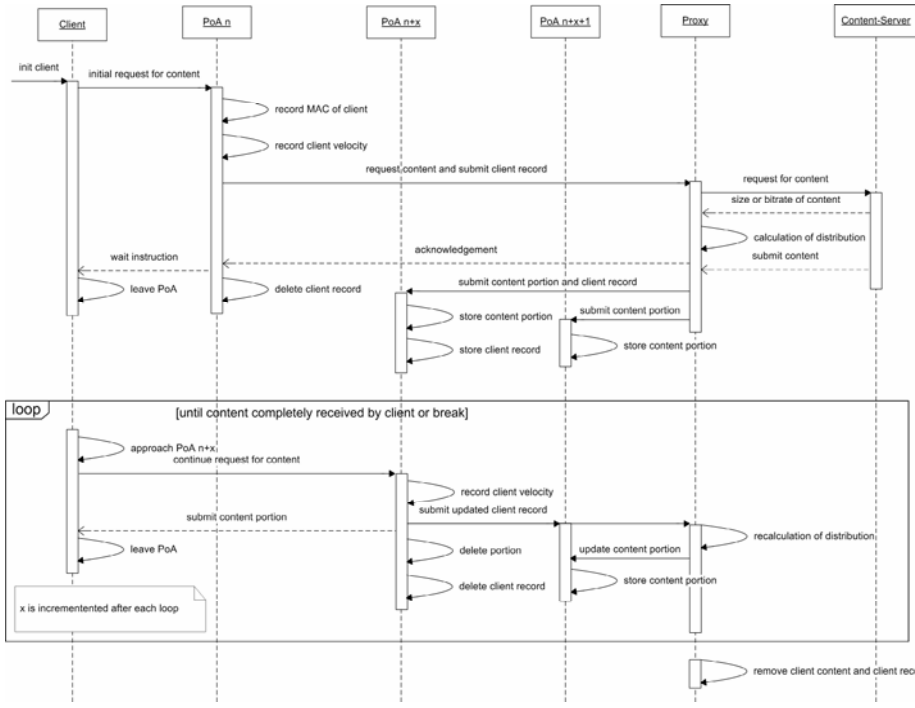


Fig. 2. Sequence diagram of our feed forward mechanism for multilayered distributed proxy system

After an initial waiting time the client can start its requested download from a PoA. The client will pass each PoA and receive CPs locally from the PoAs' caches. The client record hold by the PoA is updated with the current velocity and then transmitted to the succeeding PoA and the proxy server. After a CP was received successfully the PoA removes the client associated data records from its memory. When a CP arrives after the client at a PoA, the CP will be forwarded immediately to the succeeding PoA. After a client finished a download operation the proxy can erase all client associated data.

5 Simulation

The communication is the basis of the offered network architecture. It is necessary to investigate the dependency between the dimension of a cluster (k) and the packet propagation delay. Furthermore, the seam of two different wireless technologies (WiMAX and WLAN) produces a bottleneck. To achieve these goals a simulation model using OPNET modeler version 12.0.A PL5 was built [7]. The simplified model includes PoAs modeled with WLAN routers (the number is varied from 2 up to 10), a WWAN connection modeled with a Ethernet connection with a bandwidth of 10 MBit/s, a application server providing VoIP, and a varied number of clients (refer to figure 3). The PoAs models (refer to figure 4) use 3 WLAN interfaces: One for the

client connection and 2 for the connection to the neighboring PoAs. A standard router was extended with a channel assignment algorithm [8]. This model represents a communication cluster with ad-hoc connections. The following scenario was realized: VoIP (PCM quality speech) and FTP (one request per second with 500,000 bytes) application traffic.

The growing of the cluster dimension does not lead to relevant increasing of the average packet delivery times in case of non maximum traffic intensity. With very high traffic loads the used routing protocol (optimized link state routing, OLSR) provides a non-acceptable performance.

The simulation could show that the bottleneck on the seam WiMAX/WLAN limits the cluster dimension. The link utilization was often measured as 100 %, where the capacity of the WLAN was no limiting factor for the application profile.

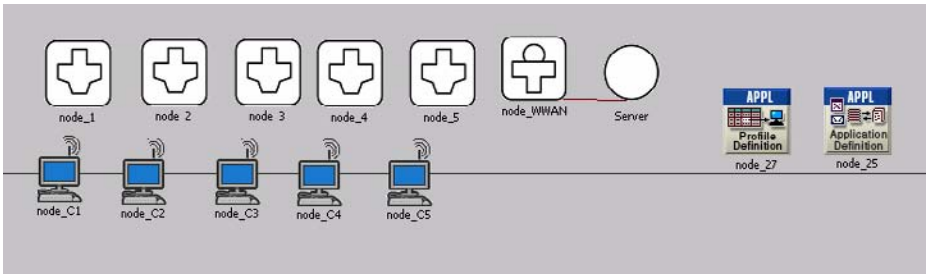


Fig. 3. Scenario with 5 PoAs and WWAN bandwidth of 10 MBit/s

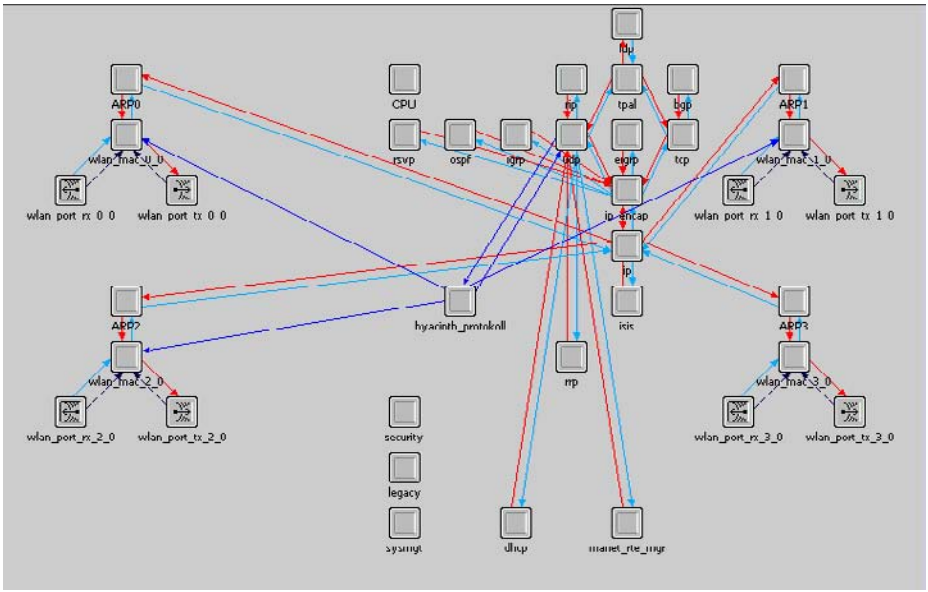


Fig. 4. Node model of a PoA

6 Conclusion and Outlook

The presented paper introduced a multi-layered architecture based on WiFi and WiMAX nodes supporting broad spectrum of applications in the highway environment. As an important feature of the offered network architecture the self-organizing and principals of pervasive communications was introduced. The system based on communication clusters which are built up automatically by the PoAs, starting at the cluster heads (process of cauterization like component of self-organizing). The realization of this kind of system is not possible only under using standard configurations and requires the extension of data management protocols. In conclusion, the feed-forward mechanism was introduced. The feed-forward mechanism realized under using of multilayered distributed proxy system and extended set of communication protocols.

There is a plenty of work to be accomplished. It is necessary to implement the feed-forward mechanism consisting of a protocol and a proxy. The possibility to improve the efficiency of the network system will be investigated in the future. More measurements for several WLAN parameters would facilitate a better understanding of the scenario and the definition of more exact system parameters to improve the system model [9].

References

[1] IEEE. 802.11a standard (1999), http://standards.ieee.org/getieee802/download/802.11a-1999.pdf

[2] IEEE. 802.11g standard (2003), http://standards.ieee.org/getieee802/download/802.11g-2003.pdf

[3] IEEE. 802.16 standard (2004), http://standards.ieee.org/getieee802/download/802.16-2004.pdf

[4] Rottenau, T., Schmeling, K., Stütz, B.G.: Kommunikation (fast) ohne Grenzen. WiMAX Performance Measurements (2006), http://www.networkcomputing.de/nwc/mobile-und-wireless/artikel/archive/180/article/kommunikation-fast-ohne-grenzen/

[5] Alois Ferscha Thematic Group 1: Pervasive Computing and Communications, Report for Public Consultation (2006)

[6] TS 102 024-2: Definition of Speech Quality of Service (QoS) Classes, http://portal.etsi.org/docbox/EC_Files/EC_Files/

[7] Website of OPNET technologies, http://www.opnet.com

[8] Raniwala, A., Chiueh, T.-c.: Architecture and algorithms for an 802.11-based multi-channel wireless mesh network. In: INFOCOM 2005, 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE, vol. 3, pp. 2223–2234 (2005)

[9] Krohn, M., Unger, H., Tavangarian, D.: Advanced Wireless Network Architectures for Highways. In: Proceedings of Wireless Congress 2007 (2007)

New Routing Strategies for RSP Problems with Concave Cost

Marjan Momtazpour and Pejman Khadivi

Department of Electrical and Computer Engineering
Isfahan University of Technology
Isfahan, Iran
{momtazpour,pkhadivi}@ec.iut.ac.ir

Abstract. Multi-Constraint Path (MCP) and Restricted Shortest Path (RSP) are important problems studied in the field of QoS routing. Traditional versions of these problems are known to be NP-Complete ones. Various solutions have been proposed for RSP and MCP based on different heuristics, in practical situations. Restricted shortest path problem with concave route costs is studied in this paper. This is a special version of the traditional RSP problem and is widely applicable in wireless and mobile ad hoc networks. In this paper, we propose new algorithms for this kind of routing. The effectiveness and performance of our proposed solutions are shown through simulations.

Keywords: Quality of Service, Routing, RSP, Ad Hoc Networking.

1 Introduction

In recent years, there has been an increasing demand for Internet-based multimedia applications. In response to this demand, the research community has been extensively investigating several quality of service (QoS)-based networking frameworks [1][2]. Routing is one of the most basic and widely studied problems in computer networking. Selecting feasible paths that satisfy various requirements of the applications running over a network, is known as Quality of Service (QoS) routing [3][4][5].

Multi-Constraint Path (MCP) and Restricted Shortest Path (RSP) problems are two well-known problems, studied in the field of QoS routing. In the RSP problem, it is required to find a feasible path that has a minimum cost among other feasible ones between a source node and a destination [4][5][6]. A feasible path is the one that satisfies a certain constraint. Routing with respect to multiple (additive) QoS requirements is known as an NP-Complete problem [7].

Khadivi et al. introduced a special version of the well-known RSP problem in [6], which is named as RSP with concave route cost (RSP-CC). In this problem, the cost of a path is the maximum of the cost of the links on that path. It is mentioned in [6] that the RSP-CC is not an NP-Complete problem. However, it is widely applicable in wireless ad hoc networks, where the distance between two neighboring nodes on a route can be considered as the link cost [6][8].

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 412–418, 2008.

In this paper, we propose new routing strategies for RSP-CC problem. While the optimum solution has been proposed in [6], it is shown that its time complexity is dramatically high. Therefore, sub-optimal methods are required in order to reduce the time required to find a feasible route. In this paper, a number of new sub-optimal solutions are proposed. The proposed strategies are evaluated through simulations.

The remainder of the paper is organized as follows. In Section 2, a detailed definition of the problem is presented. In Section 3 Pruning Based routing strategies are introduced. Section 4 proposed our new solutions. Simulation results are presented in Section 5. Concluding remarks are given in Section 6.

2 Network Model and Problem Formulation

A network can be represented by a directed or undirected graph G = (V, E), where V is the set of the nodes and E is the set of the links. Also, assume that each link $(i, j) \in E$ has an additive non-negative QoS weight, $w(i, j)$ and a concave non-negative cost, $c(i, j)$. Given a constraint, L, the Restricted Shortest Path with Concave Costs (RSP-CC) problem is to find a path $p \in \pi$ from a source node, S, to a destination node, D, with the minimum concave cost, such that,

$$w(p) \hat{=} \sum_{(i,j)\in p} w(i, j) \le L . \tag{1}$$

The concave cost of the path p, $C_C(p)$, is defined as follows:

$$C_C(p) = \max_{(i,j)\in p} c(i, j) . \tag{2}$$

A path $p \in \pi$ in an RSP-CC problem is *optimally feasible* if and only if, it has the minimum concave cost among the paths which satisfy the constraint mentioned in (1). It is shown in [6] that RSP-CC problems are not NP-Complete. However, they are widely applicable in ad hoc networks where reducing the distance between nodes on a route, results in long life routes which consumes less power [6][8][9].

3 Pruning Based Routing

In this section, an optimum and *polynomial time* algorithm is introduced which may be used for solving the RSP-CC problem [6]. The Optimum Pruning Based Routing algorithm is an iterative routing method which works based on Dijkstra. At the beginning of each iteration, the Dijkstra algorithm finds a shortest path between the source node, S, and the destination node, D. The shortest path is determined based on the additive weights. Therefore, if the result of Dijkstra is the path p^*, we have:

$$W(p^*) \le W(p) \quad ; \quad \forall p \in \pi . \tag{3}$$

where π is the set of the existing paths between the nodes S and D. If π is empty, it means that there is no path available between S and D and hence, $p^* = NULL$. If π is not empty and p^* satisfies the required constraint, L, the algorithm continues. In that case, the next step starts which is named as pruning. In this phase, the links of the network with the cost greater than or equal to the concave cost of the path, p^*, are removed. In other words, we have:

$$E = E - \{e\} \quad iff \quad c(e) \geq C_C\left(p^*\right). \tag{4}$$

where, E is the set of links in the network, e is a link, $c(e)$ is the cost of the link and $C_C\left(p^*\right)$ is the concave cost of p^*. The routing process continues until there is no path available between the nodes S and D which satisfies the required constraint. The optimality of the Optimum Pruning Based Routing algorithm is proved in [6].

While the *optimum-pruning-based-routing* algorithm, returns optimally-feasible routes, it requires a long execution time to converge to the optimum result. In the modified version of the algorithm, links are pruned based on the following rule:

$$E = E - \{e\} \quad iff \quad c(e) \geq \varepsilon \times C_C\left(p^*\right). \tag{5}$$

where, ε is a constant coefficient, $0 \leq \varepsilon \leq 1$, and we name it as the *pruning factor*. The role of this coefficient is to reach a balance between the execution time of the routing algorithm and the resulting costs. The best value which may be used for the pruning factor depends on the network's topology as well as the links' weights and costs. This modified algorithm finds an acceptable route in an acceptable time. In the rest of the paper, this modified version of the algorithm is named as Simple Modified Pruning Based (SMPB) algorithm.

4 Sub-optimum Solutions

In this section, a number of sub-optimum routing methods are proposed for RSP-CC problem. While the SMPB algorithm reduces the routing time and results in sub-optimal routes, the employed pruning factor, ε, is static and is determined off-line. Simulation results in [6] show that the value of ε has a great effect on the performance of the routing process. Hence, a dynamic value for this factor may result in better routes, respecting the route concave cost.

The first dynamic version of the modified pruning based routing, which we name it as Dynamic Factor Modified Pruning Based (DFMPB) algorithm, changes the value of ε based on $\varepsilon = (\varepsilon + 1)/2$. Therefore, with each iteration of the algorithm, ε is increased. It is almost obvious that larger values of ε result in better routes. Hence, improved performance is expected with DFMPB routing algorithm.

The second approach is the general case of DFMPB and hence is named as Generalized DFMPB. In this strategy, ε is changed based on $\varepsilon = \alpha\varepsilon + (1-\alpha)$, where, $0 \leq \alpha \leq 1$ is a constant.

```
Algorithm Simple-Bisection-Pruning;
Begin
01) TemporaryE ← E;
02) SavedRoute ← NULL
03) Route ← Dijkstra (G,S,D);
04) if (Route<>NULL)and(WEIGHT(Route)<=L) then
05) begin
06)   Search ←SearchCount;
07)   SavedRoute ← Route;
08)   PathConcaveCost ← COST(Route);
09)   L ← 0;
10)   U ← 1;
11) end
12) else
13)   Search ← 0;
14) While (Search <> 0) do
15) begin
16)   ε ← (L+U)/2;
17)   for all the links e∈ E do
18)     if (COST(e)>=ε*PathConcaveCost) then
19)       G ← G – {e};
20)   Route ← Dijkstra (G,S,D);
21)   if (Route<>NULL)and(WEIGHT(Route)<=L)
then
22)   begin
23)     SavedRoute ← Route;
24)     PathConcaveCost ← COST(Route);
25)     U ← (L+U)/2;
26)   end
27)   else
28)     L← (L+U)/2;
29)   E ← TemporaryE;
30)   --Search;
31) end
32) Return SavedRoute;
33) End.
```

Fig. 1. Pseudo-code of the Simple-Bisection-Pruning-based routing algorithm

```
Algorithm Improved-Bisection-Pruning;
Begin
01) NewLen = R; OldLen = R;
02) Search=1;
03) TemporaryE ← E;
04) SavedRoute ← NULL
05) repeat
06)   Route ← Dijkstra (G,S,D);
07)   if (Route<>NULL)and(WEIGHT(Route)<=L)
then
08)   begin
09)     OldLen = NewLen;
10)     SavedRoute ← Route;
11)     PathConcaveCost ← COST(Route);
12)     for all the links e∈ E do
13)    if(COST(e)>=ε*PathConcaveCost)then
14)         G ← G – {e};
15)     NewLen=PruningFactor*PathConcaveCost;
16)   end
17)   Else
18)   Begin
19)     E ← TemporaryE;
20)     for all the links e∈ E do
21)       if(COST(e)>=(NewLen+OldLen)/2)then
22)         G ← G – {e};
23)     NewLen=(NewLen + OldLen)/2;
24)     NumberOfFail--;
25)     if (NumberOfFail = 0) then
26)       search=0;
27)   end
28) until (search = 0);
29) Return SavedRoute;
   30) End.
```

Fig. 2. Pseudo-code of the Improved-Bisection-Pruning-based routing algorithm

Bisection is a well-known numerical method which is employed for solving linear and nonlinear equations. The same idea may be used in the case of RSP-CC routing problem. In Simple Bisection Pruning Based (SBPB) algorithm (Figure 1), two new parameters, L and U, are employed as a lower bound and an upper bound for the pruning factor, ε. After a successful routing iteration, U is updated based on $U = (L+U)/2$ while after a failed iteration we have $L = (L+U)/2$.

The routing process is iterated for a certain number of times. In Figure 1, this is defined in line 6, as Search-Count. Greater values for Search-Count result in lower concave costs, but it takes longer to find an answer. Simulation results show that the SBPB algorithm has a low performance.

The Improved Bisection Pruning Based (IBPB) algorithm, illustrated in Figure 2, combines the SMPB routing with the SBPB idea. The algorithm starts based on the SMPB. The routing process goes on based on SMPB, until a fail point is reached. Fail point is a situation that a routing iteration fails. At this point, the algorithm behaves almost similar to SBPB method, where a number of links are returned into the network. This process, which we name it as *Inverse-Pruning*, is performed in lines 20-22 of the algorithm of Figure 2.

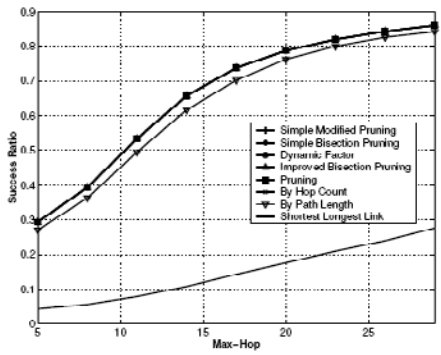


Fig. 3. Success ratio when $H_{\max}$ is changing

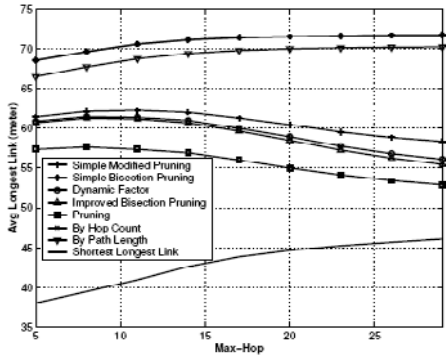


Fig. 4. Average cost when $H_{\max}$ is changing

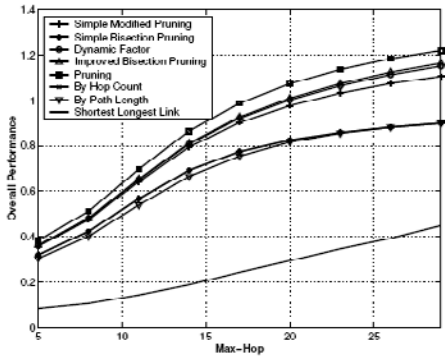


Fig. 5. Overall performance when $H_{\max}$ is changing

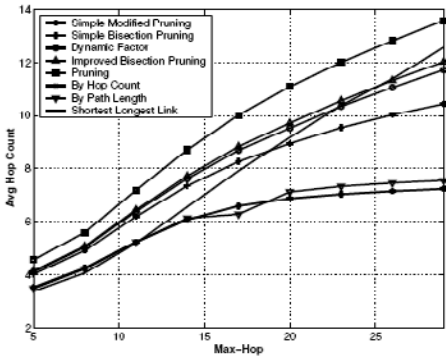


Fig. 6. Average hop-count when $H_{\max}$ is changing

5 Simulation Results

A large variety of simulation experiments have been performed using a wide range of different parameter settings. In this section, some representative results are presented which illustrate the relative performance of the proposed algorithms. In the results to be presented, two performance parameters are measured for evaluation of the proposed strategies: *Success Ratio* (SR), which is defined as the *percentage of time that the algorithm finds a feasible path*, and *average concave cost* of the generated routes.

The simulation environment is constructed by an 800×800 rectangular simulation area and 420 nodes, uniformly distributed over the area. It is assumed that there is a link between nodes *x* and *y*, if and only if, their distance is less than or equal to 75. The concave cost of a link is equal to the distance between the corresponding nodes. Also, the additive weight of the links is equal to 1. In each simulation case, the results are measured for 5000 random distributions of the nodes. The source and destination nodes are selected randomly based on the uniform distribution. For each routing case,

a constraint on the number of hops must be satisfied. This constraint is selected randomly from the range $[10, H_{\max}]$.

Similar to [6] and for comparisons, three simple routing strategies are also employed: routing based on hop-count, path-length and longest-link metrics. Path-length is the summation of the length of links in the path. With the longest-link metric, we try to find a path with the shortest longest link.

The routing algorithm is successful, if and only if, the required constraint is satisfied. In Figure 3, different routing strategies are compared from the success ratio (SR) point of view. It is clear from Figure 3 that the proposed solutions have the same SR as the routing based on the hop-count metric. Also, it is obvious that routing based on the *Shortest-Longest-Link*, results in unacceptable success ratios.

Different routing strategies are compared in Figures 4 with respect to the average concave cost of the routes. It is clear from this figure that routing based on the shortest-longest-link metric result in the lowest costs. However, the behavior of this strategy is not acceptable, because its SR is very low. The proposed optimum-pruning-based solution, results in acceptable SR and cost values. Routing based on Hop-Count and Simple Bisection Pruning method, results in almost similar concave costs. Simulation results show that among modified pruning solutions, Improved Bisection (IBPB) is the best method.

In order to have a clear view of the performance of the proposed algorithms, let us define the following performance metric:

$$Overall \quad Performanc\text{œ} = \left(\frac{Success \quad Ratio}{Average \quad Cost} \right) \times 75 \cdot \tag{6}$$

"Success Ratio" and "Average Concave Cost" are both taken into account when "*Overall Performance*" is used in the comparisons. A routing algorithm behaves well, when its success ratio is high and the average cost of the routes is low. Higher success ratios and lower average costs result in higher "overall performance". The results are illustrated in Figure 5. It is clear from this figure that the proposed optimum-pruning-based strategy, results in the highest *overall-performances*. Again, Improved Bisection (IBPB) is the best candidate among the modified pruning solutions. The average hop-count of the routes under different solutions is illustrated in Figure 6.

6 Conclusions

In this paper, a special version of the well-known RSP problem is considered, which we name it as *RSP with concave route cost*. This problem is widely applicable in wireless ad hoc networks. Previously, we have studied this problem in [6]. However, the Optimum-Pruning Based routing strategy, introduced in [6], requires a long execution time to converge to the optimum result. The modified version of this solution returns a sub-optimal route in a dramatically shorter period of time. In this paper, other solutions are proposed to find routes with lower cost in appropriate execution time. Simulation results show the effectiveness of our proposed methods, in terms of Success Ratio and average Concave Cost.

References

1. Korkmaz, T., Krunz, M.: Multi-constrained optimal path selection. In: Proc. of INFOCOM 2001, vol. 2, pp. 834–843 (2001)
2. Kuipers, F., Mieghem, P.: An Overview of Constraint-Based Path Selection Algorithms for QoS Routing. IEEE Communications Magazine, 50–55 (December 2002)
3. Wang, Z.: Internet QoS: Architectures and Mechanisms for Quality of Service. Morgan Kaufmann Publishers, San Francisco (2001)
4. Cheng, G., Ansari, N.: Achieving 100% Success Ratio in Finding the Delay Constraint Least Cost Path. In: Proceedings of IEEE GlobeCom 2004, pp. 1505–1509 (2004)
5. Khadivi, P., Samavi, S., Todd, T.D., Saidi, H.: Multi-Constraint QoS Routing Using a New Single Mixed Metric. In: Proc. of IEEE Int. Conf. on Communications (ICC 2004), France, vol. 4, pp. 2042–2046 (2004)
6. Khadivi, P., Samavi, S., Saidi, H.: Restricted Shortest Path Routing with Concave Costs. In: Proceedings of the 4th ACS/IEEE International Conference on Computer Systems and Applications, Dubai/Sharjah, UAE (2006)
7. Wang, Z., Crowcroft, J.: Quality-of-service routing for supporting multimedia applications. IEEE Journal on Selected Areas in Communications 14, 1228–1234 (1996)
8. Khadivi, P., Todd, T.D., Samavi, S., Saidi, H., Zhao, D.: Mobile Ad Hoc Relaying in Hybrid WLAN/Cellular Systems for Dropping Probability Reduction. In: Proceedings of the 9th CDMA International Conference (CiC 2004), Korea (2004)
9. Sreng, V., Yanikomeroglu, H., Falconer, D.D.: Relayer Selection Strategies in Cellular Networks with Peer-to-Peer Relaying. In: Proceedings of the IEEE VTCF 2003 (2003)

Modeling Routing Protocols in Adhoc Networks

Fatemeh Ghassemi and Ali Movaghar

Sharif University of Technology, Tehran, Iran
fghassemi@mehr.sharif.edu, movaghar@sharif.edu

Abstract. Regarding increasing popularity of Ad hoc networks, the routing protocols employed in these networks should be validated before deployment. Formal methods are used nowadays to find defects in protocols specification. In this paper, we explain different methods of formal modeling and verification of routing protocols of ad hoc networks. We derive the key concepts that are vital in modeling ad hoc network protocols and then modify process algebra, appropriate for verifying protocols at network layer. This process algebra implements mobility behaviors of underlying infrastructure implicitly in the semantics of broadcasting. The semantics of broadcast communication also abstracts away the services provided by MAC layer.

Keywords: Routing protocol, Ad hoc Networks, Process Algebra, Connectivity, Implicit Mobility, Restricted Broadcasting.

1 Introduction

According to [1] there are more than one hundred routing protocols available now and almost a new one is developed yearly. The routing protocols are simulated to be tested. As it is known simulation can not explore all conditions exist to verify such systems. Formal methods can be used to model and then verify such protocols using model checking techniques by tools automatically. In this paper we explain different methods applied to verify routing protocols in ad hoc networks.

Routing protocols can be classified into the two categories namely, reactive and proactive. Reactive routing protocols are on demand protocols, i.e., a route to a node is found when there is a need. In contrast, in proactive protocols, nodes propagate update information to their neighbors periodically. So the timing constraints should be considered in modeling and verifying such proactive protocols, while reactive protocols can be modeled by sequencing process algebra. Process algebras provide strong techniques to reason about these systems at modeling level. In this paper, we focus on verifying reactive routing protocols in ad hoc networks. Using basic processes (such as CSP[2], CCS[3] and ACP[4]) only the concurrency behaviors of processes can be modeled and the more detailed properties are abstracted away. For instance, the topology of network is considered static and the detail information about messages and their structures are not considered. As we focus on verifying routing protocols in ad hoc networks by process algebra, it is essential first to find the properties that affect on the behavior of protocols and then extend the standard process algebras with auxiliary processes, to imitate the required properties such as broadcast communication,

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 419–426, 2008.

asynchronous communication and topology changes. This approach may lead to a process algebra with lots of auxiliary processes which increases complexity at the description (modeling) level. Another disadvantage of this approach is that on the semantics level, some important aspects cannot be treated at all. For example, semantics which take into account localities cannot be used [5]. It turns out that it is more appropriate to modify the standard process algebras to model properties directly. In this work, we propose process algebra suitable for verifying reactive routing protocols in ad hoc networks. Our process algebra is based on multi-way synchronization of CSP and channel-based communication and grouping in pi-calculus.

Structure of paper: In section 2 the formal approaches applied to verify ad hoc protocols are introduced. In Section 3 the important properties that should be considered in design of algebra are explained. In Section 4 we explain our algebra, its syntax, and semantics. In Section 5, we will explain our conclusions and future works.

2 Related Works

In this Section we will explain different approaches for verifying routing protocols in ad hoc networks and we also explain about their challenges. We postpone related process algebras verifying networks (other than routing) protocols to Section 4.

SPIN [14]: In [7,8, 20] the SPIN model checker has been applied to verify Wireless Adaptive Routing Protocol (WARP) [9] and Lightweight Underlay Network Ad hoc Routing protocol [10] (LUNAR) respectively. SPIN is a model checker made for fixed communication protocols by the use of the PROMELA language. Systems are modeled as finite state machines. In [7], the challenges are modeling broadcast communication and mobility. Promela is based on point to point communication. Thus to model the broadcast system, there must be as many channels as nodes. So an array of channels is defined to enable broadcasting to any desired node. To overcome mobility, a non-determinism initialization for defining the network configuration is used, so the model checker checks all possible configurations. Due to mobility, the state space increase exponentially with the complexity of the protocol model. To verify WARP [7], five nodes were considered, however they use abstraction to derive a basic model. Abstraction was done by defining set of rules managing the basic model, such as node 1 sends only one data message to node 5 after the link updates are propagated. The link update propagations and message delivery in every five-node network respecting rules were checked.

In [8], the modeling considerations and the type of ad hoc properties that can be realistically be verified are introduced. The modeling considerations, which are not tied to the protocol itself, are connectivity, dynamics and broadcasting which are the same as ones in [7]. The property under verification is correct operation of a protocol. The correct operation is defined as follows [8]: “If there is at one point in time exists a path between two nodes, then the protocol must be able to find some path between the nodes. When a path has been found, .and for the time it stays valid, it shall be possible to send packets along the path from source node to the destination node”. The connectivity of nodes is modeled by a symmetric, two dimensional, array of Boolean values and node dynamics are modeled by modifying the connectivity matrix. Broadcasting is modeled by unicasting to all nodes with whom the sending node presently has

connectivity. An array of channels indexed by node id should be defined to enable communication (like [7]). However, using this approach, the state space is already large without any node connectivity transition. Therefore, the verification focused on a few interesting scenarios of node movements. In [11] the correct operation of LUNAR protocol route establishment is examined in realistic general scenarios by applying abstraction using UPPAAL. The abstraction implies when a message is transmitted over a link chain between two nodes, the message reaches the intended receiver using the fastest path.

UPPAAL [15]: In [8, 11, 12, 13, 19] has been applied to verify real time aspects of routing protocols as a network of timed automata, extended by data types.

In [8,11] timing requirements of LUNAR, such as theoretical lower and upper bounds on route formation and message delivery are checked. As in the SPIN, the UPPAAL model uses arrays of Booleans to represent inter-node connectivity. Node connectivity transitions are modeled using a separate automaton that at any time can move to its next state whereby it manipulates the (global) connectivity matrix. Broadcasting is handled similarly to unicasting. As mentioned before in [11] an abstraction for simplifying the model is defined.

In [12], a pattern to model mobile ad hoc networks in UPPAAL is provided, including encoding of locations and mobility as well as local broadcast where the actual receivers of messages are those nodes only that are immediate neighbors of the emitting node. A location is modeled as a set of groups (of nodes) to which a node belongs. Local broadcast and unicast in UPPAAL is modeled using broadcast channels with guards. A broadcast channels allows an automaton which has the broadcast channel as an output transition, to be synchronized with an arbitrary number of automata, that all have the same broadcast channel as an input transition.

In [13], the flooding protocol is modeled and verified. In that a logical sequence is assumed for messages as mentioned in Section 1 which reduce the model. In this work connectivity is modeled by an automaton called gridpoints which uses two-dimensional array to show if two nodes are adjacent. The sequence of messages is modeled by scheduler automaton.

The works introduced so far which focus on ad hoc networks, insist on modeling connectivity and dynamism which affect on the behavior of routing protocols itself. They also focus on modeling broadcasting as the only means of communication in ad hoc network which can be local and global broadcasting.

Process algebra: In [16], the backoff algorithm in ad hoc network is verified using PEPA, stochastic process algebra. In this approach the topology of network is static and broadcasting is implemented by unicasting. To our knowledge there is no specific process algebra modified to verify routing protocols.

Other: In [13,17] program algebra, Kleene-style algebra, is used to verify the probabilistic behavior of protocol formally using refinements. However the operational semantics of program algebra is equivalent to that of probabilistic automata used in PRISM [13]. In contrast to these works, we focus on verifying concurrency behavior of protocols in ad hoc networks and the timing constraint and probabilistic behavior of protocols are not considered, we will not consider these algebra in designing our algebra.

3 Ad Hoc Properties

We have explained different approaches and their challenges in Section 2. The properties (challenges) that should be considered are broadcasting, connectivity, and dynamism.

An ad hoc network consists of numbers of independent nodes acting in parallel and having collaboration and cooperation with each other. To this aim, nodes communicate with each other. The only means of communication in ad hoc networks is broadcasting. In general, our algebra should support restricted broadcasting. In standard algebras like CCS and CSP, communication is based on one-to-one synchronous through shared channels. In CBS, which is explained later in Section 4.1, provides unbounded broadcasting. Communication capability of nodes is defined directly by the underlying topology (connectivity). Two nodes can communicate if they are connected. A routing protocol is specified independent of underlying topology but it should behave correctly for all topologies and topology changes.

4 Routing Process Algebra

The early works on designing process calculi were focused around modeling protocol and reasoning about these models. We can mention to Lotos and its descendents. In contrast, some recent developments focused on enriching the calculus from specification aspect to be applied as the basis for a programming language. We can mention recent extensions of pi calculus such as Nomadic pi [22]. We focus on modeling and reasoning about protocols to keep the specification simple to examine concurrency behaviors regarding topology changes. In designing the Routing Process Algebra, RBA, the properties explained in Section 3 should be considered. In this section, first we explain similar existing works to support properties.

4.1 Related Process Algebra

The *Calculus of Broadcasting Systems*, CBS [6], is algebra where the only means of communication is broadcasting. In this approach broadcasting is an autonomous action and only one process can broadcast at a time. When a process broadcast, all other process can receive. During broadcast communication, a process broadcast and others receive. This is an abstract view of broadcasting. We take and modify broadcasting defined in CBS to restricted broadcasting.

Nomadic pi [24], an extension of pi calculus, is suitable for verifying distributed and mobile systems. This algebra provides a two-level framework; at lower level location-dependant primitives are introduced while at the second layer location-independent primitives are defined using underlying primitives. This algebra models communication using channels. We define broadcasting via channels exchanging data. We also take naming and group concepts from pi calculus to model connectivity.

4.2 RPA Syntax

The syntax of RPA is shown below:

$\Gamma \vDash [\![P]\!]_{A_1} \|_G [\![P]\!]_{A_2} \|_G \ldots \|_G [\![P]\!]_{A_n}$, $A_1, A_2, \ldots, A_n \in Names$

$P ::= 0 \mid a.P \mid P+P \mid P \bowtie_L P \mid P/R \mid [x \sim x]P$, $a \in \{c\ x!, c\ x?\} \cup Action$,
$c \in Channel$ and $x \in Data$ and $\sim \in \{=, >, <, <>\}$

where 0 is a deadlocked or failed process. A process failure can be result of computer crash or hardware problems, etc. The "a.p" shows a process which is able to perform an action "a" and then behaves like "p" process. The action "a" can be a broadcast/receive on a channel or one of the predefined actions of context process represented by Action. The data broadcast from a channel can have predefined Data types in process context. The variable "x" in "c x?.p" is a bounded variable in process p. The only binding operator in RPA is receiving.

The process "p_1+p_2" behaves non-deterministically like process "p_1" or "p_2". The process "$p_1 \bowtie_L p_2$" defines multi-way synchronization of two processes "p1" and "p_2" on the set of actions defined in L. We have taken this operator from PEPA to define action synchronization between processes. To encapsulate some internal behavior of a process, hide operator is used. The process "P/R" insures the action defined in "R" can not be synchronized with any other processes. The guarded command enables to make a decision on data received via a channel. The process "[x=id]" behaves like P if value of "x" is equal to "id". We have taken this from Nomadic pi to enrich our algebra from the specification aspect.

A network, $[\![p_1]\!]_{A_1} \|_{g_1} [\![p_2]\!]_{A_2} \|_{g_2} \cdots \|_{g_n} [\![p_n]\!]_{A_n}$, is defined by composition of processes p_1, p_2,…,p_n run on context A_1, A_2,…,A_n via broadcasting communication. We model connectivity by groups. Physical locations in ad hoc networks can be modeled by groups, i.e., nodes in a group are within broadcasting range of each. So group concept can be used to model physical and logical connectivity. Restricted broadcasting is defined in terms of groups. The broadcast composition operator, "||", is defined by parameter g as "||g" to define groups involving in broadcast communication. We call "g" as the broadcast zone of broadcast communication. Multicasting can be implemented by broadcasting to a group containing nodes involving in multicast communication. However we have not considered group creation and deletion in our process algebra and we fed the grouping management implicitly into the semantics. In general words, the behavior of a process at a node is defined by operational semantics underlying a topology represented by Γ which describes network connectivity. The topology Γ is a total function from Names to a powerset of Names. Each name is a representative of a group containing itself. To describe a process called "p" at node "A", we use "$[\![p]\!]_A$" terminology.

It should be noted that communication composition is only defined between processes (protocols) run at independent nodes. The inner structure of nodes can be sequence ".", choice "+", parallel synchronization "$\bowtie$", hide "/" and guarded command "[]". A protocol is an ongoing process running at a node, so we add recursion to our syntax to support infinite computation.

$$\frac{}{a.P \xrightarrow{a} P} : \textbf{Action} \qquad \frac{P_1 \xrightarrow{a} P_1'}{P_1 + P_2 \xrightarrow{a} P_1'} : \textbf{Choice} \qquad a \in R, \frac{P \xrightarrow{a} P'}{P/R \xrightarrow{\tau} P'/R} : \textbf{Res}_1 \qquad a \notin R, \frac{P \xrightarrow{a} P'}{P/R \xrightarrow{a} P'/R} : \textbf{Res}_2$$

$$a \notin L, \frac{p_1 \xrightarrow{a} p_1'}{p_1 \rhd\lhd_L p_2 \xrightarrow{a} p_1' \rhd\lhd_L p_2} : \textbf{Comp}_1 \qquad x \sim x \equiv True, \frac{}{([x \sim x]p) \to p} : \textbf{Gua}_1$$

$$a \in L, \frac{p_1 \xrightarrow{a} p_1' \quad p_2 \xrightarrow{a} p_2'}{p_1 \rhd\lhd_L p_2 \xrightarrow{a} p_1' \rhd\lhd_L p_2'} : \textbf{Comp}_2 \qquad x \sim x \equiv false, \frac{}{([x \sim x]p) \to 0} : \textbf{Gua}_2$$

Fig. 1. Operational semantics of NBA

4.3 RPA Semantics

Before explaining semantics of operators, the following rule holds:

$$[\![P \to P']\!]_A \equiv [\![P]\!]_A \to [\![P']\!]_A$$

We will use structural operational semantics introduced by plotkin to define the formal semantics of RPA operations. Each consists of three parts: pre-condition, SOS rule and post-condition. Pre and post condition are specified by first order logic. The formal semantics of sequence, choice, hide and parallel synchronization are straightforward shown in Fig. 1. Parallel synchronization for actions out of between them previously. This rule, called Bro_2 in this context, models link breakage implicitly. On the other hand, two nodes participate in a broadcast communication if a connection link has been recently created between them. This rule, called Bro_3, models link creation implicitly. Parallel synchronization for actions out of synchronization set has interleaving and these actions can be performed independently. The action "τ" is a hidden action representing intra communications between inner processes of a process (protocol). When a hide operator acts on a process, it filters actions defined in *restriction set* "R". As shown in Fig.1, a guarded process proceeds if the guard satisfies to true, otherwise the process is failed (a deadlock will happen as the behavior of system is not defined for other conditions).

$$B \in \Gamma(A) \subseteq G, \frac{[\![P_1 \xrightarrow{cx!} P_1']\!]_A \quad [\![P_2 \xrightarrow{cy?} P_2']\!]_B}{[\![P_1]\!]_A \|_G [\![P_2]\!]_B \xrightarrow{cx!} [\![P_1']\!]_A \|_G [\![P_2'[x/y]]\!]_B} : \textbf{Bro}_1 \tag{1}$$

$$A \in G, \frac{[\![P_1 \xrightarrow{cx!} P_1']\!]_A \quad [\![P_2 \xrightarrow{cy?} P_2']\!]_B}{[\![P_1]\!]_A \|_G [\![P_2]\!]_B \xrightarrow{cx!} [\![P_1']\!]_A \|_G [\![P_2]\!]_B}, B \notin \Gamma(A) : \textbf{Bro}_2 \tag{2}$$

$$B \notin \Gamma(A) \subseteq G, \frac{[\![P_1 \xrightarrow{cx!} P_1']\!]_A \quad [\![P_2 \xrightarrow{cy?} P_2']\!]_B}{[\![P_1]\!]_A \|_G [\![P_2]\!]_B \xrightarrow{cx!} [\![P_1']\!]_A \|_G [\![P_2'[x/y]]\!]_B}, B \in \Gamma(A) : \textbf{Bro}_3 \tag{3}$$

$$A,B \in G, \frac{[\![P_1 \xrightarrow{cx!} P_1']\!]_A \quad [\![P_2 \xrightarrow{cx!} P_2']\!]_B}{[\![P_1]\!]_A \|_G [\![P_2]\!]_B \xrightarrow{cx!} [\![P_1']\!]_A \|_G [\![P_2]\!]_B} : \textbf{Bro}_4 \tag{4}$$

Fig. 2. The operational semantics of broadcast communication

The operational semantics of operators explained in Fig. 1 irrespective of the context they are running at (They hold for any context). The operational semantics for broadcast communication is shown in Fig. 2. Rule 1 in Fig. 2 implies two processes

communicate if they belong to a same group (or location). After the communication, the message is forwarded to be received by other processes included in group "G" and monitoring channel "c". The default value of broadcast zone is the transmitter group. In contrast to CBS, we define interleaving for broadcasting to implement mobility implicitly. The rule 2 in Fig 2. indicates two processes do not communicate in a broadcasting, if the link between them has recently broken or there was no connection

The rule 4 in Fig.2 indicates only one process can broadcast on a channel non-deterministically. As we focus on network layer, the collision management in MAC layer should be abstracted away. Thus the abstract view of collision management is that only one process can broadcast at the time. We can conclude two processes can broadcast simultaneously if they broadcast on different channels. The interleaving rule, called Bro_4, also abstracts away the hidden problem as the semantics does not allow two processes on different groups broadcast simultaneously. This rule also supports Input Blocking/Output non-Blocking behavior as specified as a required behavior for protocols in [21].

5 Conclusion and Future Works

In this paper, we modify new process algebra, called Routing Process Algebra (RPA), based on CSP multi-way synchronization for intra synchronous communication, channel based communication and grouping of pi calculus and broadcast of CBS.

In design of RPA, we focus on network layer and properties of ah hoc protocols. The network layer uses the services provided by underneath layer, MAC layer, so the collision management should be abstracted away in this layer. We have abstract away of such problems in our semantics. The only means of communication in ah a hoc network is broadcasting. Behavior of protocol running at each node is dependent to the underlying infrastructure. Thus to verify a routing protocol we have modeled underlying infrastructure using group concepts. Grouping abstracts unimportant moves when there is no change in node connectivity. We have also implemented mobility behavior of nodes implicitly to verify behavior of protocol against topology changes. This implementation is satisfied by semantics of broadcasting. We have added restriction zone to the broadcast of CBS and also have added Bro_2, Bro_3 and Bro_4 to its semantics to support mobility and collision free communication.

The advantages of our process algebra to previous ones are its capability in modeling restricted broadcasting considering node connectivity and implementing mobility changes implicitly.

We are going to examine reasoning at algebra level to simplify the model formally. Abstraction can be defined by finding a simulation relation between two sub-protocols. We are going to examine bisimulation relation between protocols to define composability and substitutability for routing protocols as specified in [21].

References

1. List of ad-hoc routing protocols (last visited 27, October 2007), `http://en.wikipedia.org/wiki/List_of_ad-hoc_routing_protocols`
2. Hoare, C.A.R.: Communicating Sequential Processes. Prentice-Hall International, Englewood Cliffs (1985)

3. Milner, R.: Communication and Concurrency. Prentice-Hall International, New York (1985)
4. Bergstra, J.A., Klop, J.W.: Algebra of Communicating Processes with Abstraction. Theor. Comput. Sci. 37, 21–77 (1985)
5. Gruska, D.P., Maggiolo-Schettini, A.: Process Algebras for Network Communication. Fundamenta Informaticae 45(4), 359–378 (2001)
6. Prasad, K.V.S.: A Calculus of Broadcasting Systems. Journal of Science of Computer Programming 25(2-3), 285–327 (1995)
7. de Renesse, R., Aghvani, A.H.: Formal verification of Ad-Hoc Routing Protocols using SPIN Model Checker. In: Proceedings MELECON 2004, Dubrovnik. IEEE Press, Los Alamitos (2004)
8. Wibling, O., Parrow, J., Pears, A.: Automatized Verification of Ad Hoc Routing Protocols. In: de Frutos-Escrig, D., Núñez, M. (eds.) FORTE 2004. LNCS, vol. 3235. Springer, Heidelberg (2004)
9. Khengar, P., Aghvami, A.H.: Wrap- a new hybrid routing protocol for mobile ad hoc networks. IEEE Journal on Selected Area in Communications (2004)
10. Tschudin, C., Gold, R., Rensfelt, O., Wibling, O.: LUNAR:a lightweight underlay network ad-hoc routing protocol and implementation. In: Proc. Next Generation Teletraffic and Wired/Wireless Advanced Networking (NEW2AN) (2004)
11. Wibling, O., Parrow, J., Pears, A.: Ad Hoc Routing Protocol Verification Through Broadcast Abstraction. In: Wang, F. (ed.) FORTE 2005. LNCS, vol. 3731, pp. 128–142. Springer, Heidelberg (2005)
12. Godskesen, J.C., Gryn, O.: Modeling and Verification of Security Protocols for Ad Hoc Networks Using UPPAAl. In: Proceeding of the 18th Nordic Workshop on Programming Theory (NWPT 2006), Iceland, October 18-20 (2006)
13. McIver, A.K., Fehnker, A.: Formal Techniques for Analysis of Wireless Network. In: Margaria, T., Philippou, A., Steffen, B. (eds.) Proc. 2nd Int. Symp. ISOLA (2006)
14. Holzmann, G.: The SPIN Model Checker, Primer and Reference Manual. Addison-Wesley, Reading (2003)
15. Larsen, K.G., Pettersson, P., Yi, W.: Uppaal in a Nutshell. Int. Journal on Software Tools for Technology Transfer 1, 134–152 (1997)
16. Razafindarlambo, T., Valois, F.: Performance Evaluation of Backoff algorithms in 802.11 AdHoc Networks. In: Mobile Computing and Networking, pp. 48–57 (2002)
17. McIver, A.K., Cohen, E., Morgan, C.C.: Using Probabilistic Kleene Algebra for Protocol Verification. In: Schmidt, R.A. (ed.) RelMiCS/AKA 2006. LNCS, vol. 4136, pp. 296–310. Springer, Heidelberg (2006)
18. Patsouris, P.A.: Algebraic modeling of an ad Hoc network for mobile computing. Journal of Parallel Distribution and Computing 61(7), 884–897 (2001)
19. Chiyangwa, S., Kwiatkowska, M.: A Timing Analysis of AODV. In: Steffen, M., Zavattaro, G. (eds.) FMOODS 2005. LNCS, vol. 3535, pp. 306–321. Springer, Heidelberg (2005)
20. Bhargavan, K., Obradovid, D., Gunter, C.A.: Formal Verification of Standards for Distance Vector Routing Protocols. Journal of the ACM 49(4), 538–576 (2002)
21. Niamanesh, M., Jalili, R.: Formalizing Compatibility and Substitutability in Communication Protocols Using I/O-Constraint Automata. In: Arbab, F., Sirjani, M. (eds.) FSEN 2007. LNCS, vol. 4767, pp. 49–64. Springer, Heidelberg (2007)
22. Sewell, P., Wojciechowski, P., Pierce, B.: Location Independence for Mobile Agents. In: ICCL-WS 1998. LNCS, vol. 1686. Springer, Heidelberg (1999)

CEBAC: A Decentralized Cooperation Enforcement Based Access Control Framework in MANETs*

Fatemeh Saremi, Hoda Mashayekhi, Ali Movaghar, and Rasool Jalili

Department of Computer Engineering, Sharif University of Technology, Tehran, Iran
{f_saremi,mashayekhi}@ce.sharif.edu,
{movaghar,jalili}@sharif.edu

Abstract. Prevention of unauthorized access to services in mobile ad hoc networks is a more sophisticated problem than access control in other networks, due to interconnection facilities and lack of any fixed network infrastructure in such networks. Therefore regarding the nature of these networks, controlling access to services should be in a decentralized manner providing good performance and preserving network security. In this paper, we propose a decentralized Cooperation Enforcement Based Access Control (CEBAC) framework for mobile ad hoc networks. CEBAC comprises several groups of Service Authorizers, each issuing Credentials for access to a specific kind of services. The User Authorization for using services, besides the internal possibly traditional access control model of Service Providers, is governed by the specific threshold-based access control scheme introduced in CEBAC. In addition, a punitive mechanism is applied to mitigate the selfish behavior of authorizers and motivate them to cooperate.

Keywords: Cooperation, Group Management, Access Control.

1 Introduction

Access control is a major security aspect in ad hoc networks which consists of the means to govern the way the users or virtual users can have access to services. Existence of no fixed network infrastructure makes deciding about users' access a very time- and power-consuming task to Service Providers. So in this paper we propose a framework with a group of Service Authorizers selected by each Service Provider for this purpose. Because of limited power, Service Authorizers specifically, may tend to not respond users' requests. Doing so causes the process of acquiring Credentials to be very time-consuming as well as wasting much power of the network. To overcome this problem, CEBAC uses a punitive mechanism to motivate authorizers to cooperate.

LHAP [3] is an authentication protocol that emphasizes on resource consuming attacks. For prevention of these attacks, it performs node to node authentication. LHAP consists on reducing cryptographic operations because of their computational overhead.

* This research was in part supported by a grant from Iran Telecommunication Research Center.The draft version of this work can be found in: http://ce.sharif.edu/~f_saremi/ CEBAC.pdf

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 427–434, 2008.

The methods would not efficiently work if access to a service should be controlled. ID_GAC [8] is an identity based membership control technique for ad hoc groups. It is based on the threshold variant [9] of BLS signature scheme [10] and also uses Verifiable Secret Sharing in bootstrap phase. RAMARS [11] proposes a role-based access management framework to enable secure resource sharing. The framework incorporates role-based approach to address distributed access control, delegation and dissemination control involved in the resource sharing within ad hoc collaboration environments. The remainder of this paper proceeds as follows. We introduce CEBAC framework briefly in section 2. In section 3, we define the framework's model formally. Architecture and communication flow of CEBAC are described in section 4. After a discussion of the framework in section 5, section 6 concludes the paper with future research directions.

2 CEBAC Overview

2.1 A CEBAC Scenario

Suppose that different kinds of services are to be provided in an ad hoc network. We focus on a particular service s and a user u requesting to access s. Instead of having a centralized service provider which authorizes the users and presents them sub-services and objects to access, we propose a Service Group which consists of Service Providers (SPs) all providing service s, and Service Authorizers (SAs) which authorize users and issue needed credentials for them. Credential is a signed certificate which indicates rights and privileges to access a service. SAs are chosen and given the required information by the SPs using an appropriate method. In a service group there may be one or more SPs and at least t SAs. t is a parameter for key management and credential issuing discussed later in 2.2. SPs and SAs of a particular service altogether form an ad hoc group and some required shared secrets are maintained among its members. One important secret is the private key K of the group and each member keeps a part of it namely a partial key k_i. K could be constructed with interpolation if at least t partial keys are present. This way the private key is not revealed to outsiders even if up to t-1 group members are compromised. A corresponding public key is available to all nodes in the network. Mechanisms of group key management in a service group are discussed in detail in 2.2. Each SP provides the users with some services and related objects which are documented in the SP's attributes.

User u has a set of attributes and credentials which are used to gain access credentials for services. When u intends to use s, it issues a request to a local SA and submits the required credentials and attributes. SA verifies the credentials, and other things such as the nodes behavior history. After verification of the required material which we call Service Access Prerequisites, SA issues a credential indicating u has the required permissions to access the service. The credential issued for the user is signed by the partial key owned by the SA, and it will be referred in this context as a Partial Credential. The creation time of the Partial Credential is embedded in it as a Time Stamp. This time stamp is return to the user in plain text too. After gaining the first Partial Credential, the user must obtain t-1 other ones from t-1 other SAs and he can gain them in parallel. The only difference is that besides submitting the required credentials and attributes, the user must also submit the initial Time Stamp each time,

so that all the other SAs issue Partial Credentials with the same time stamp. This is required to resist some tricks performed by malicious nodes to gain unauthorized access. SAs check the Time Stamp against current time and the difference should not exceed an indicated parameter. u combines the t partial credentials to obtain the main credential signed by the private key K of the group [5]. This way the authorization is done in a distributed manner, so resisting many security attacks and the extra workload is assigned to user. u can now access the service or objects provided by the SP. u submits a message containing the service credential and the requested access mode to SP. SP verifies the credential by decrypting it with group public key, then allows or denies access according to the policy it has for access control of its service or objects and the requested access mode.

In the procedure described above some SAs may behave selfishly and respond wrongly or not respond at all. u's device can keep a list of such authorizers and after gaining access to the service, report the list to SP. SP records these complaints and according to a special policy, say by a threshold on the number of reports, punishes the selfish SA. SP must be aware of reports from malicious users who submit wrong or repetitive reports. The kind of punishment is up to SP. A proposed one is isolating the selfish node in the network. In the case of multiple SPs in a group, they must share their knowledge of reports on selfish nodes in regular time intervals.

2.2 Service Group Administration

Each service group has policies for administering and managing the group related activities which are chosen by the SP or administrator of the service. Selection of the group members is up to the SP and any policy based on the credentials and attributes of the nodes can be used. Each group has a private key which is used to sign credentials assigned to users. Because of security vulnerabilities in ad hoc networks, it is critical to keep the key secret. Many previous works have proposed efficient group key management schemes. In [13] authors use threshold secret sharing to establish a secret group key, and each member keeps a part of it, namely a partial key. By obtaining t partial keys, the group secret key can be computed through interpolation. t is a parameter that is selected as a trade off of performance and security.

The partial keys should be updated in regular intervals to decrease the chance of malicious nodes in obtaining the key. In threshold cryptography schemes for issuing certificates, with t partial shares of the key k, one could obtain k with interpolation. Also if t contributors sign message m with their partial keys of key k, then with interpolation one could obtain message m signed by k with no extra knowledge other than possibly the interpolation function and parameters [5]. After obtaining t partial credentials, which are analogous to t partial signatures, u combines them to obtain the main credential.

2.3 Reporting Selfish Behavior

In the procedure of gaining partial credentials by the user, some SAs might decide not to reply due to selfish behavior. If user records these misbehaviors, after gaining access to service, it can report them to SP. SP after checking the identity of u and verifying that the reports are not repetitive, accepts and keeps it. In regular intervals, which

can be the same as update intervals of the group partial keys mentioned in 2.2, these reports are shared between SPs. Each SP after aggregating the reports it has on each SA, and comparing the result to a predefined threshold, can decide which SA should be punished. Any other suitable scheme specifying selfish SAs can be used by SPs. The network nodes are informed not to forward the selfish node's packets.

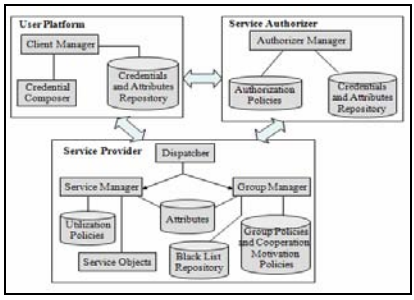


Fig. 1. The architecture of the CEBAC

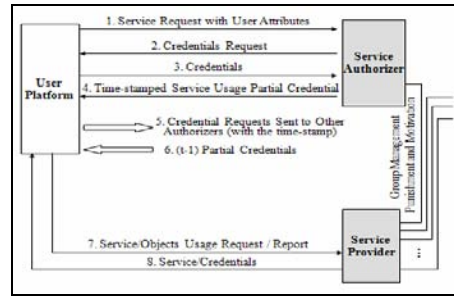


Fig. 2. The CEBAC's communication sequence

3 System Model

The CEBAC model consists of 5 basic elements.

Definition 1. The general model for the CEBAC is a 5-tuple (U, SG, AM, ServPlc, SysAttr), in which U is the set of Users of the system, SG is the Service Group which provides the service and performs authentication and authorization, AM is the Access Modes available for accessing the service, ServPlc is Service Policies and authorization rules, and SysAttr is the System Attributes.

Definition 2. The Service Group is a 3-tuple (SP, SA, GP), where SP is the Service Provider who provides accesses to objects, SA is the Service Authorizer who checks the prerequisites and issues the needed credentials for accessing a service, and GP is the Group Policies and management considerations for the Service Group.

Definition 3. Group Policies in the SG entity is a set defined as union of 4 policies.

GP = Group Administration Policies ⋃ Membership Management Policies ⋃ Key Management Policies ⋃ Black List Management Policies.

Definition 4. Access Modes in the general system model is a 2-tuple (GAM, SAM), where GAM is the General Access Modes common to all services and SAM is Specific Access Modes which are particular to each service.

Definition 5. Service Policy in the system model is a set defined as follows:

ServPlc = ServAcsPreReq ⋃ ServAcsActiveReq ⋃ ObjAcsPreReq ⋃ ServAcsPostReq in which ServAcsPreReq is the prerequisites for accessing the service, ServAcsActiveReq is the active requisites which should be satisfied during the access to service and violating any of them results in revoking the ongoing access, ObjAcsPreReq is prerequisites for accessing any sub-service or object after gaining the credentials

for accessing the service, and ServAcsPost-Req is the requisites of ending the ongoing access which generally causes changes in credentials of the user.

Definition 6. ServAcsPreReq is a set defined as follows:

ServAcsPreReq=SystemAttributes ∪ UserCredentials ∪ UserAttributes ∪ SAAttributes.

Definition 7. ServAcsActiveReq is a set defined as follows:

ServAcsActiveReq=SystemAttributes ∪ UserCredentials ∪ UserAttributs ∪ SPAttributs.

4 The Architecture

4.1 Architecture Components

Fig. 1 shows the architecture of CEBAC. The architecture encompasses three main components: Service Provider (SP), Service Authorizer (SA), and User Platform(UP).

User Platform. UP component is composed of Client Manager and Credential Composer modules and Credentials and Attributes Repository. Client Manager has the responsibility of sending requests to SAs and collecting t partial credentials, handling and retrieving user's attributes and credentials, service access, and handing over the reports to SP. The other module, Credentials Composer, combines partial credentials and computes a valid service usage credential with an embedded time-stamp in it. Client Manager stores the new credential in Credentials and Attributes Repository. The credential remains in the repository until expiration of its Time Stamp.

Service Authorizer. SA component includes an Authorizer Manager module and two repositories: Authorization Policies and Credentials and Attributes Repository. Authorizer Manager verifies the prerequisites of service access, issues and exports service usage's credentials to users according to its credentials, Authorization Policies, maintains users' attributes and credentials, and service usage prerequisites. Each SA holds its credentials, attributes, and service authorization's partial key in its Credentials an attributes Repository.

Service Provider. SP component comprises Dispatcher, Service Manager, Service Objects, and Group Manager modules and four repositories: Utilization Policies, Attributes, Group Policies, and Black-List Repository.

Dispatcher scans the received messages and according to their type, delivers them to Service Manager module or Group Manager module. Service Manager performs service usage related affairs. It receives user's credentials and decides upon user's access rights on Service Objects based on Utilization Policies, user credentials, and Service Objects' Attributes, Object Access Prerequisites and Service Active access Requisites. Service Manager is responsible for taking into account the SP's internal authorization rules. Group Manager chooses a group of trusted nodes as SA. This group is established based on Group Policies and the nodes' attributes and credentials. Then key management affairs are performed in the group. Group Manager also receives the users' reports about the selfish SAs and keeps information on selfish authorizers in Black-List Repository. Based on Group Policies it can choose to punish an authorizer.

4.2 Communication Flow

Any user u who wants to use service s should acquire a credential. To do this, u should collect at least t partial credential and combine them to acquire a valid service usage credential. Figure 2 shows this procedure in brief details.

Partial Credentials Collection. In order to use a service, the service usage's permission is needed. In previous MANETs' access control models permission is usually acquired via one authorizer. So they are vulnerable to attacks, and if one of the authorizers be compromised, unauthorized accesses to services could be easily done. Therefore in order to mitigate this problem, we make the service usage dependent on acquiring at least t partial credentials from t distinct authorizers. So the network will be secure against compromising t or less than t authorizers.

For the purpose of collecting t partial credentials, in accordance with CEBAC, u hands over its attributes besides its request to an authorizer. Determining the value of threshold t is an engineering problem. On the one hand if t has a large value, the security of network increases, while smaller t makes the network more vulnerable. On the other hand, a large value of t decreases the probability of acquiring permission to use the service. Its value can be the same everywhere in the network, or it can be location-dependent. The latter gives more flexibility, but as the distribution of authorizers is not known as a priori, and there is no clear system-wide trust assumption in MANETs, we use the former option. To obtain t partial credentials, u sends its required attributes and credentials to a SA in its neighborhood and repeats until one authorizer issue it to gain a time-stamped partial credential. In the next steps u sends the acquired time-stamp besides the required attributes and credentials to at least t-1 other SAs in its neighborhood in parallel.

Issuing Partial Credential. When an authorizer receives a u's service request, it retrieves the list of prerequisites from Authorization Policies Repository. After informing u of the prerequisite list and receiving them, the authorizer issues a partial credential respecting Authorization Policies with an embedded time-stamp. If u's service request is associated with a time-stamp TS, the authorizer after checking some conditions on TS, embeds it in the partial credential issued for u. It is worth mentioning that the authorizer signs the issued credential using its partial key.

Credential Computation and Service Access. After acquiring t partial credentials successfully, u can combine them to compute a service usage credential (via Credential Composer module). This new credential is stored in u's Credentials and Attributes Repository.

Possessing a valid credential (with a valid time-stamp) u can access the desired service/objects. Also the access is governed by SP's internal access control model. In addition some requisites (ServAcsActiveReq) should be held during service access, or else u will be prohibited from service access continuation. After service access, some credentials (ServAcsPostReq) may be issued by SP to u. u stores these credentials in its repository. Selfish behavior of authorizers is a common problem in MANETs that the user might face in collecting t partial credentials. Authorizers may not respond the requests of users, with the aim of saving their power. The probability of this condition happening is significant considering the limited power of nodes in MANETs,. To overcome this problem, we provide the possibility of reporting SAs' selfish behavior to SPs for the users. When a user succeeds to access SP, it can report the authorizers' misbehavior.

Group Selfishness Treatment. When a user reports an authorizer's selfishness, SP reconsiders the authorizer's membership in the Service Group. SP treats the authorizer based on Group Policies and puts the authorizer in its Black-List. Then it informs the other SPs. Through this punitive mechanism the probability of SAs' selfish behavior decreases and users can access services in shorter times. This mechanism decreases network's power consumption, because user is not forced to communicate with much more than t authorizers and authorizers' communication load will decrease.

5 Discussion

CEBAC assumes a group of nodes with a hierarchical structure, providing a particular kind of service. This way, safer nodes are assigned specifically for providing service and richer, more expensive schemes are utilized for their security, while nodes in lower levels have the responsibility of authorizing users to access the service. When the number of nodes in the network is incremented, owing the scalability of CEBAC, increasing the SAs will preserve performance. The distributed authorization process we proposed prevents malicious nodes to access the service in an unauthorized manner. Being fully distributed, the proposed authorization mechanism improves security of the network. Collecting t partial credentials enforcement, leaves colluding at least t malicious nodes in a same time interval $T_{key\text{-}upd}$ as the only way to unauthorized service usage. So the network will be highly attack resistant. Besides, when the number of expelled authorizers in same time interval exceeds t, we might update partial keys to prevent them collude. In addition, for the sake of confidentiality and integrity, all the messages are encrypted with shared or public/private keys. Due to presence of timestamp in partial credentials a malicious node can not combine new and old partial credentials to obtain a valid credential.

Concerning limited power of nodes, CEBAC distributes channels' communication traffic in a fair way through dividing and spreading the responsibilities among different nodes. Saving the computational power of the nodes is the other advantage obtained. Separating the responsibility of authorization and assigning it to authorizer nodes, a user can gain the required credentials for accessing the service almost locally. Besides, we win the benefit of providing the service to more users. To meet scalable, dynamic, and fine-grained authorization requirements, SPs have the flexibility of choosing any required access control scheme for protecting their internal objects.

CEBAC is user centric, say it is the user who is responsible for keeping, combining or presenting its credentials when necessary. This quality also increases security because no central repository is present for credential storage, so attackers have much less chance to achieve confidential information and no heavy traffic is produced for credential retrieval. In addition to checking users' id and certificates, by considering their attributes and credentials during service usage continuously, we take the context into account of authorization (dynamic authorization).

Controlling selfish behavior of nodes and their lack of adherence to network operations in the special case of authorizers is another important aspect of CEBAC that has been hidden in the view of available approaches. The mechanism operates through providing users the ability to report misbehavior of authorizers to SPs, and then facing guilty authorizers based on each SP's punitive policies. This brings many advantages to CEBAC framework: it enforces authorizers not to behave selfishly, the process of authorization will be more efficient, users' and authorizers' power is better reserved,

and a locally known selfish authorizer will become a globally known selfish one, because the other SPs are also informed about the guilty authorizer. In CEBAC framework, a SP selects its group members via a trust-based scheme. The trust-based scheme can be completed by additional specifications of recommendations and past experience of users. Therefore SPs can use the reports prepared by users to do so. The scheme detail is not specified in this paper and will be illustrated in extensions.

6 Conclusion and Future Work

In this paper, we proposed a decentralized Cooperation-Enforcement Based Access Control framework for mobile ad hoc networks. We applied a special distributed scheme in which special nodes collaboratively authorize users. In addition, we provided the capability of monitoring selfish behavior of authorizers. We expressed the formal definition of our model and briefly presented the communication flow and architecture of the main functional modules. Substantial work remains to be done towards the trust-based selection of the authorizers-group's members by SPs. Finally, we intend to evaluate our framework by means of a real world implementation.

References

1. Marti, S., Giuli, T.J., Lai, K., Baker, M.: Mitigating Routing Misbehavior in Mobile Ad Hoc Networks. In: MOBICOM 2000 (2000)
2. Zhou, Y., Zhang, Y., Fang, Y.: Access control in wireless sensor networks. Ad hoc Networks 5, 3–13 (2007)
3. Zhu, S., Xu, S., Setia, S., Jajodia, S.: LHAP: A lightweight network access control protocol for ad hoc networks. Ad hoc Networks 4 (2006)
4. Wan, Z., Deng, R.H., Bao, F., Ananda, A.L.: Access control protocols with two-layer architecture for wireless networks. Computer Networks 51 (2007)
5. Zhou, L., Haas, Z.J.: Securing ad hoc networks. IEEE Network 13, 24–30 (1999)
6. Zhang, X., Nakae, M., Covington, M.J., Sandhu, R.: A Usage-based Authorization Framework for Collaborative Computing Systems. In: SACMAT 2006, June 7–9 (2006)
7. Luo, H., Kong, J., Zerfos, P., Lu, S., Zhang, L.: URSA: Ubiquitous and Robust Access Control for Mobile Ad hoc Networks. IEEE/ACM transactions on networking 12(6), 1049–1063 (2004)
8. Saxena, N., Tsudik, G., Yi, J.H.: Identity-based Access Control for Ad hoc Groups (2004)
9. Boldyreva, A.: Efficient threshold signatures, multisignatures and blind signatures based on the gap-diffie-hellman-group signature scheme. In: International Workshop on Practice and Theory in Public Key Cryptography. LNCS, vol. 2567, pp. 31–46. Springer, Heidelberg (2003)
10. Boneh, D., Lynn, B., Shacham, H.: Short Signatures from the Weil Pairing. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 514–532. Springer, Heidelberg (2001)
11. Jin, J., Ahn, G.J.: Role-based Access Management for Ad hoc Collaborative Sharing. In: SACMAT 2006, June 7–9 (2006)
12. Keoh, S.L., Lupu, E.: Towards Flexible Credential Verification in Mobile Ad hoc Networks. In: POMC 2002, October 30–31 (2002)
13. Castelluccia, C., Saxena, N., Yi, J.H.: Robust self-keying mobile ad hoc networks. Computer Networks (2006)

A Secure Cross-Layer Design of Clustering–Based Routing Protocol for MANET

Arash Dana and Marzieh Hajhosseini

Iran Telecommunication Research Center,
Central Tehran Branch Islamic Azad University

Abstract. Most of routing protocols proposed for ad hoc networks have a flat structure. These protocols expand the control overhead packets to discover or maintain a route. On the other hand a number of hierarchical-based routing protocols have been developed, mostly are based on layered design. These protocols improve network performances especially when the network size grows up since details about remote portion of network can be handled in an aggregate manner. Although, there is another approach to design a protocol called cross-layer design. Using this approach information can exchange between different layer of protocol stack, result in optimizing network performances. In this paper, we intend to exert cross-layer design to optimize Cluster Based Routing Protocol (Cross-CBRP). Using NS-2 network simulator we evaluate rate of cluster-head changes, throughput and packet delivery ratio. Comparisons denote that Cross-CBRP has better performances with respect to the original CBRP.

Keywords: Cross-layer design, Cross-CBRP, Cluster-head election.

1 Introduction

When utilizing a communication infrastructure is expensive or impossible, mobile users can still communicate with each other through a wireless ad hoc network. Because of limited radio range of mobile nodes a packet is constrained to traverse several hops. Moreover, the mobility of nodes combined with transient nature of wireless links cause network topology changing. Because of these issues a number of routing protocol with different structures created; flat routing protocols and hierarchical routing protocols. In an ad hoc network with flat routing protocol all nodes have the same role in packet forwarding. Therefore protocol performances degrade when the network size increases. In hierarchical routing protocol like fewer nodes have outstanding role in packet routing and other nodes role is inconspicuous.

CBRP is a routing protocol that has a hierarchical-based design [7], [9]. This protocol divides the network area into several smaller areas called cluster. The clustering algorithm of CBRP is Least Cluster Change or LCC [10] means the node with the lowest ID among its neighbors elects as cluster-head. Because of mobility of nodes in ad hoc network this is probable that elected cluster-head to be too mobile. In addition, because nodes with cluster-head role consume more power than ordinary nodes, mobile node with lower ID discharge soon. Through these reasons cluster-head election procedure used in CBRP is not suitable.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 435–442, 2008.

We used cross-layer design to solve this problem. In this structure each layer is characterized by some parameters. These parameters then passed to adjacent layers to help them adapt themselves for best suit the current channel, network, and applications [8].

To realization this approach, signal strength was used to determine mobility of nodes. This parameter is shared between Phy, MAC and network layers to achieve a better cluster-head election algorithm. In fact we used cross-layer approach to elect an appropriate node as cluster-head to reduction of cluster-head changes rate and therefore superior protocol performances.

2 Related Works

A number of clustering algorithms have been proposed in literatures that create clusters that their maximum diameter can be two or more hops. Linked Clustered Algorithm (LCA) [1], Lowest-ID (LID) [2], Maximum Connectivity (MCC) [3] and Least Cluster Change (LCC) are the most famous traditional algorithms. Most of these algorithms have a simple random criterion to elect a cluster-head mainly focuses on how to form clusters with a good geographic distribution, such as minimum cluster overlap, etc. These kind of clustering algorithms don't meet stability of clusters; however, it is an important criterion especially when clustering used to support routing. To meet this end some other clustering algorithms was created that considered cluster stability. A number of this kind of clustering algorithms can find in [4], [5], [6], [12] and some other literatures. In [12] cluster-head election parameter is node's mobility. In [4] cluster-head election is based on mobility and power quantity of nodes. In [5], [6] a weight-based clustering algorithm is proposed. Collection of mobility, link connectivity, power and distance of nodes are gathered to elect a cluster-head. The advantage of these algorithms is their precise criterion in cluster-head election and therefore more stable clusters creation.

In this paper we focused on the cluster-head election algorithm of CBRP. We used cross-layer approach to elect cluster-heads for it. We replaced MOBIC [12] clustering algorithm instead of its original algorithm means LCC.

3 Cross Layer Approach for CBRP: Cross-CBRP

Mobile Ad hoc networks experience severe topology changes in addition to common problems of other wireless networks. Successive join-and-leave nature of MANET nodes in hierarchical algorithms like CBRP, that is fully dependent on the cluster-heads behavior, directly influences the overall network performance. Therefore, wise cluster formation as a mainstream part of these algorithms can improve network performance. In CBRP, cluster formation is performed with a simple and naive approach of the lowest ID. In such a raw selection, every node with the lowest ID between its local neighbors will be cluster head. Obviously, neither the network dynamics nor the clusters stability has been considered. As mentioned earlier, in hierarchical cluster-based MANET, cluster-heads play the main role in maintaining the cluster structure and standing against the destructive factors namely mobility. In the cross layer design

approach proposed in this paper, cluster formation mechanism and cluster maintenance are considered with respect to proportional mobility of the node towards its neighbors. With this scheme, a node with the lowest mobility and movement in the pre-specified period of time will be named cluster-head. By means of cluster-head stabilization, network will not suffer from cluster tumbling and local destruction in addition to overheads caused by that.

Recent experimental studies have demonstrated that as the availability of links fluctuates because of channel fading phenomena, the effects of the impairments of the wireless channel on higher-layer protocols are not negligible. Furthermore, mobility of nodes is not considered. In fact, due to node mobility and node join-and-leave events, the network may be subject to frequent topological reconfigurations. Thus, links and clusters are continuously established and broken. This process in hierarchical cluster-based architecture will result in excessive overhead and cluster-head change which degrades performance of the whole network. For the above reasons, new analytical parameters and information from link layer are required to help network layer to determine connectivity conditions; containing mobility and fading channels. In our new approach, the sense of network dynamics and topography changes in physical layer (in the form of received signal power) is fully exploited in network layer cluster formation to achieve energy efficiency and robustness against topological dynamicity [11].

3.1 An Aggregate Local Mobility for Cross-CBRP

We use Rayleigh fading model to describe the channel between wireless nodes in a cluster. For a transmitter-receiver separation x, the channel gain is given by:

$$h(x) = L(d_0)\left(\frac{x}{d_0}\right)^{-n}\xi. \tag{1}$$

where $L(d_0) = G_t G_r l^2/16\pi^2 d_0^2$ is the path loss of the close-in distance d_0, G_t is the antenna gain of the transmitter, G_r is the antenna gain of the receiver, l is the wavelength of the carrier frequency, n is the path loss exponent ($2 \le n \le 6$), and ξ is a normalized random variable that represents the power gain of the fading. Using equation (1), will give us $P_r/P_t \propto x^{-n}\xi$, and by neglecting randomness of fading effect we will have,

$$P_r / P_t \propto x^{-n}. \tag{2}$$

The equation (2) shows an inverse n-th power dependence of the radio of received and transmitted power on the physical distance between the transmitter and the receiver. In reasonably short time scales e.g. a few seconds, the surrounding environment is unlikely to change significantly, therefore the variable channel gain caused by the effects of multipath, small-scale and large scale fading can be ignored. In this situation the variation of the received signal power will be a good indicator for local mobility of every node.

The ratio of P_r between two successive packet transmissions i.e. periodic “hello” messages from a neighboring node will get us a good knowledge about the relative

mobility between two nodes. From this the relative mobility metric $M_Y^{rel}(X)$ at a node Y with respect to X can be define as:

$$M_Y^{rel}(X) = 10 \log_{10}\left(\frac{p^{new}_{r_{X\to Y}}}{p^{old}_{r_{X\to Y}}}\right). \quad (3)$$

Now consider a node with *m* neighbors; there will exist *m* such values for $M_Y^{rel}(X)$. We use the aggregate local mobility value M_Y at any node Y by calculating the variance (with respect to zero) of the entire set of relative mobility samples $M_Y^{rel}(X_i)$, where X_i is a neighbor of Y as proposed in [12]:

$$M_Y = \mathrm{var}\{M_Y^{rel}(X_i)\}_{i=1}^{m} = E[(M_Y^{rel})^2]. \quad (4)$$

In this paper $M_Y^{rel}(X_i)$, in (3) is used as a mobility characteristic of a node with respect to its neighbors. As it can be seen from (4) every node is able to calculate M_Y, just from a comparison between received powers of "hello" packets in the successive periods of time. Aggregate local mobility of nodes will be included in the advertising packets and broadcasted to neighbors in addition to the node ID and other CBRP's common fields. Resorting to this new field of information, each node makes a table which keeps the track of two parameter for every neighbor; ID and aggregate local mobility. During the cluster formation algorithm, when eligible nodes are competing for taking cluster-head role in a distributed manner, aggregate local mobility of every node computed formerly by advertised "hello" packets is compared with aggregate local mobility of its neighbors. For the sake of maximum stability in this heuristic topology control algorithm, the node with the lowest aggregate local mobility will win and take the cluster-head role. For better adaptation to uncommon circumstances, lowest ID will be considered just in the rare condition of mobility metric equality. Here, it should be highlighted that M_Y and power estimations which are the building blocks of determinant tables in addition to the tables themselves are gathered, processed and stored locally just with the aid of neighbor's "hello" packets. Despite the fact that each node computes its mobility just with respect to neighbor's contributions independently (an indirect approach), there is no need to have a central node to collect and redistribute node's information with a lot of overhead which means scalability in a mobile Ad hoc network.

3.2 Distributed Cluster Formation Algorithm for Cross-CBRP

In order to use the aggregate mobility metric presented in the section 3.1 for clustering, we propose a two step distributed clustering algorithm which use the mobility metric as a basis for cluster formation. You can find the description of the algorithm in the following paragraph:

All nodes send (receive) "Hello" messages to (from) their neighbors. Each node measures the received power levels of two successive transmissions from each neighbor, and then calculates the pair wise relative mobility metrics using (3). Also, every node extracts the relative mobility metric of every neighbor from received "hello" packet. Then, each node computes the aggregate relative mobility metric M_Y using (4). All nodes start in Cluster-Undecided state. Every node broadcasts its own

mobility metric, M_Y (initialized to 0 at the beginning of operation) in a "hello" message to its 1-hop neighbors, once in every Broadcast-Interval (BI) period. If this node is not already in the neighbor table of each neighboring node, will be stored in the neighbor table of them along with a time-out period (TP) seconds as a new neighbor. Otherwise neighboring node situation becomes update. This algorithm is distributed. Thus, a node receives the M_Y-values from its neighbors, and then compares them with its own. If a node has the lowest value of M_Y amongst all its neighbors, it assumes the status of a cluster-head. Then this node broadcasts a "hello" packet to introduce itself as cluster-head. In case where the mobility metric of two cluster-head nodes is the same, and they are in competition to retain the cluster-head status, then the selection of the cluster-head is based on the Lowest ID algorithm in which the node with lowest ID gets the status of the cluster-head. If a node with cluster member status and with low mobility moves into the range of another cluster-head node with higher mobility, re-clustering will not triggered (similar to LCC [10]) because this is in contrary to the network stability and overhead mitigation.

4 Results and Discussions

The simulations were performed using the ns-2 network simulator with the MANET extensions [13]. The mobility scenarios were randomly generated using the random waypoint mobility model with 100 nodes randomly distributed in a 1000 m × 1000 m. The transmission range of the nodes considered 250 m and scenarios was generated such that nodes be always mobile (P.T=0). The maximum speed of nodes have been considered 10, 20, 30 m/sec. Traffic is generated using NS-2 CBR traffic generator. There are simultaneously 60 CBR traffic flows associated with randomly selected disjoint source and destination nodes. Packet size is set to 512 bytes. We used DropTail/PreQueue for implementing the interface queue. The size of the queue buffer sets to 50. We implemented Cross-CBRP by doing the required modification on the latest implementation of CBRP in ns-2 environment [14]. Two kinds of scenarios were used to evaluate the network performances. Each simulation has been run for 300 seconds, and the results are averaged over 5 randomly generated nodal spatial topologies. We precisely compared performance parameters of our proposed approach with the original CBRP such as rate of cluster-head changes, throughput and packet delivery ratio. Throughput is defined as the average number of data packets received at destinations during simulation time and packet delivery ratio is defined as the total number of data packets sent by traffic sources to the total number of data packets received at destinations.

In the first scenario, the rate of packet sent is 4 byte per seconds. Max mobility speed has been considered 10, 20 and 30 m/sec. Fig. 1 shows the effect of varying mobility on the performance of Cross-CBRP with respect to CBRP. It can be seen explicitly that Cross-CBRP outperforms CBRP by averagely 37% improvement for cluster-head changes. It is very clear that Cross-CBRP yields a remarkable gain over CBRP because of its capability of adapting itself to the mobility of nodes. From cluster-head changes vs. mobility curve, we can conclude that Cross-CBRP is suitable for stable cluster formation in situations involving mobility. Fig. 2(a) demonstrates the packet delivery ratio differences of two algorithms in the existence of mobility. Again we can see that in average the Cross-CBRP performs about 9% better than

CBRP because of the cross-layer adaptation technique that has been used in its design. The throughput plays an important role in comparing different network protocols from QoS perspective. Fig. 2(b) demonstrates the results of measured throughput for two previously discussed protocols. The performance results show more efficient behavior of Cross-CBRP in comparison with CBRP with respect to mobility. As it is apparent from the Fig.2 (b), the Cross-CBRP outperforms CBRP about 8.5% which again supports this claim that increasing cluster stability we will give us better network performance.

In the second scenario we changed the sent packet rate from 1 pkt/sec to 8 pkts/sec. In this scenario we intend to study effect of varying traffic on the performance of Cross-CBRP with respect to original-CBRP. As shown in Fig. 3 rate of cluster-head change increases with the packet rate augmented. When traffic injection to network increases some reasons can cause packets do not received by downstream node for example lack of route or impossibility to access to the media – so packets will hold in the interface queue. If this buffer overflows the last incoming packet will discard. Therefore, if there are some hello packet in this queue these hello packets reach to the neighbors nodes by delay. Two cluster-head may have a uni-directional link with each other in this elapsed time; so both of them remain as cluster-head until their link changes to bi-directional link. When hello messages reach to destination uni-directional link can change to bi-directional. Therefore, one of adjacent cluster-heads must change its role. This will cause the cluster-head changing rate increase by increasing traffic injection to network. As we seen in this figure the rate of cluster-head

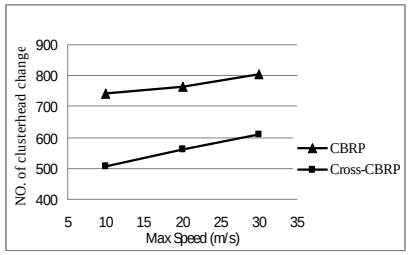
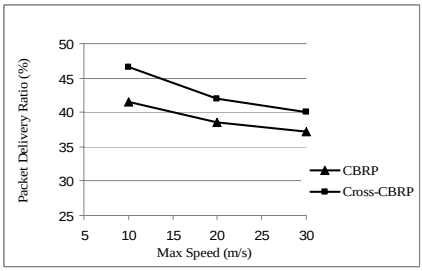


Fig. 1. Number of Cluster-head Changes vs. Speed



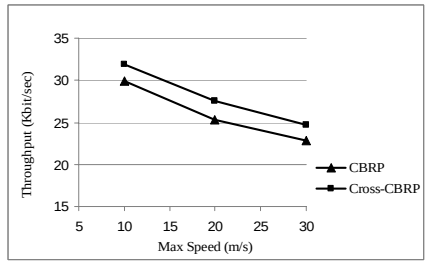


Fig. 2. Comparison between CBRP and Cross-CBRP (a) Packet delivery ratio vs. Speed (b) Throughput vs. Speed

changes in Cross-CBRP is out perform original CBRP about 30% in low packet rate and 10% in a high packet rate. This is again because of Cross-CBRP capability to adapting itself to the network conditions.

Fig. 4 again illustrates the packet delivery ratio and throughput of network versus packet rate. Traffic injection to the network causes degrading probability of access to the media. Whatever, the injected packet to network increases, packet delivery ratio decreases. We can see in Fig. 4(a) that in average the Cross-CBRP performs about 9% better than CBRP because of the cross-layer adaptation technique that has been used in its design. Fig. 4(b) shows the results of measured throughput for two previously discussed protocols. Although packet delivery ratio decreases when traffic rate increases, increasing throughput continued. This is because of increasing amount of injected traffic in the network that cause number of received packet bytes increases. It can be seen from Fig. 4(b) that Cross-CBRP throughput outperforms original-CBRP about 10% when traffic increases.

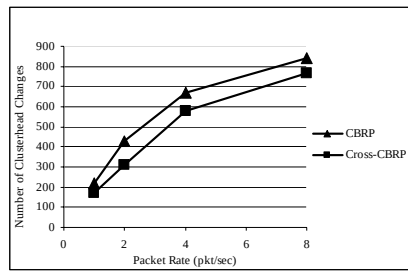
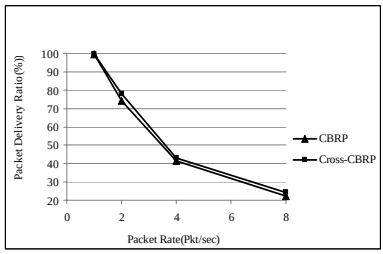


Fig. 3. Number of Cluster-head Changes vs. Packet Rate



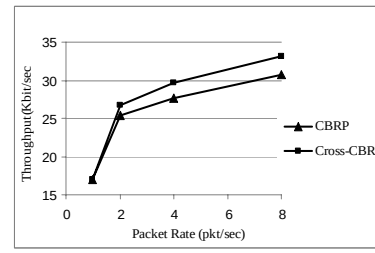


Fig. 4. Comparison between CBRP and Cross-CBRP (a) Packet delivery ratio vs. Speed (b) Throughput vs. Speed 6

Clustering algorithms as discussed in section 3 provide more efficient way to utilize the network resources like bandwidth and energy. Mobility of nodes in MANET has a destructive role in the efficient resource allocation. In this paper, we presented a new approach to cross-layer design of CBRP to enhance its efficiency with respect to the existence of mobility in Ad hoc networks. Cross-CBRP, by considering multiple layers such as physical, MAC and network layer tries to provide an adaptive clustering algorithm. Using ns-2 we demonstrated that Cross-CBRP outperforms CBRP in

different performance factors. Therefore, we conclude that Cross-CBRP, using mobility parameter sensed via physical layer, is able to behave much better than CBRP which does not account for mobility issues at all. We believe that the cross-layer approach for designing clustering protocol for Ad hoc and wireless sensor networks is a productive field of research. It is possible, to account for other parameters from the physical layer such as channel state to provide more reliable adaptive clustering protocols regarding varying behavior of wireless channels like fading and noise effects.

References

1. Baker, D.J., Ephremides, A.: The architectural organization of a mobile radio network via a distributed algorithm. IEEE Transactions on Communications 29(11), 1694–1701 (1981)
2. Ephremides, A., Wieselthier, J., Baker, D.: A design concept for reliable mobile radio network with frequency hopping signaling. Proceedings of IEEE 75, 56–73 (1987)
3. Parekh, A.K.: Selecting routers in ad hoc wireless networks. In: ITS (1994)
4. Er, I.I., Seah, W.K.G.: Mobility Based d-Hop Clustering Algorithm for Mobile Ad hoc Networks. In: Proc. IEEE Wireless Communications and Networking Conference (March 2004)
5. Purtoosi, R., Taheri, H., Mohammadi, A., Froozan, F.: A light- weight Contention-Based Clustering Algorithm for Wireless Ad hoc Networks. In: Proc. Forth International Conference on Computer and Information Technology, IEEE, Los Alamitos (2004)
6. Chatterjee, M., Das, S.K., Turgut, D.: An on-demand weighted clustering algorithm (WCA) for Ad hoc networks. In: Global Telecommunications Conference, 2000. GLOBECOM apos, vol. 3, pp. 1697–1701. IEEE, Los Alamitos (2000)
7. Jiang, M., Li, J., Tay, Y.C.: Cluster Based Routing Protocol. IETF Draft (August 1999)
8. Setton, E., Yoo, T., Zhu, X., Goldsmith, A., Girod, B.: Cross layer design of ad hoc network for real time video streaming. IEEE wireless communications, 59–65 (August 2005)
9. Mingliang, J., Tay, Y.C., Long, P.: CBRP: A Cluster-based Routing Protocol for Mobile Ad hoc Networks, National University of Singapore, http://www.comp.nus.edu.sg/~tayyc/cbrp/hon.ppt
10. Chiang, C.C., Wu, H.K., Liu, W., Gerla, M.: Routing in clustered multi-hop mobile wireless networks with fading channel. In: Proceedings of IEEE Singapore International Conference on Networks (SICON 1997), pp. 197–211 (1997)
11. Melodia, T., Vuran, M.C., Pompili, D.: The State of the Art in Cross-Layer Design for Wireless Sensor Networks. In: Cesana, M., Fratta, L. (eds.) Euro-NGI 2005. LNCS, vol. 3883, Springer, Heidelberg (2006)
12. Basu, P., Khan, N., Little, T.D.C.: A Mobility Based Metric for Clustering in Mobile Ad hoc Networks. In: Proc. IEEE ICDCSW 21st International Conference on Distributed Computing Systems Workshops (April 2001)
13. Wireless and Mobility Extensions to the ns-2 Network Simulator, CMU Monarch Project, http://monarch.cs.cmu.edu/cmu-ns.html
14. Network Simulator - ns-2. UC Berkeley, http://www-mash.cs.berkeley.edu/ns/

An Approach for Semantic Web Query Approximation Based on Domain Knowledge and User Preferences

Zeinab Iranmanesh, Razieh Piri, and Hassan Abolhassani

Department of Computer Engineering,
Sharif University of Technology, Tehran, Iran
{iranmanesh@ce,piri@ce,abolhassani@}sharif.edu

Abstract. One of the most important services in the Semantic Web is the Reasoning Service. According to the Semantic Web requirements, reasoning under time pressure or other restrictions is needed; and, reasoning which is not 'perfect' but instead 'good enough' for given tasks is acceptable. One of the approaches for the improvement of reasoning performance is approximation; of course, there is an effort for raising more precise approximations. One of the fields in which approximation seems to be useful is query. So far, all of the approximation strategies introduced for conjunctive queries just consider the query's structure. In this paper, a semantic approach for conjunctive query approximation utilizing domain knowledge and user preferences is proposed. Like other applications, it is expected that the usage of the suggested approach, either singular or in combination with other syntactic methods, results in a substantial improvement in the final approximation.

Keywords: Semantic Web, Reasoning, Reasoning Approximation, Description Logic, Object-Oriented Modeling.

1 Introduction

One of the primal challenges concerning reasoning services is the interest for using them in Semantic Web. Description logic is a suitable tool to represent ontologies and also for reasoning on them. For making the reasoning process more scalable and efficient, approximation can be used in different parts of a reasoning system as the query. In this paper a heuristic approach is introduced for semantic web query approximation, especially conjunctive queries, using the knowledge in the ontologies which are presented as TBox and ABox (Description Logic Knowledge Base); in fact the approach is more semantic in comparison to the so far proposed methods for conjunctive query approximation. In section 2, related works are discussed. In section 3, a method for query approximation is proposed and in section 4 the conclusion is provided.

2 Related Works

2.1 Fuzzy RDF and OWL

In [1], combining fuzzy logic principles with RDF data structure is studied. Because the relevance property in fuzzy logic, a query language is chosen which contains

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 443–452, 2008.

aggregation, min, max functions and those functions which provide ordering and data types. The solution is to choose an OWL query language and enhance it but the problem is that in languages like OWL-QL, DQL, SAIQL, these important features are not proven to exist.

2.2 OWL Full Reasoning from an Object Oriented Perspective

An OWL processor has been developed called SWCLOS on top of CLOS (Common Lisp Object System) to bridging the gap between OWL and object oriented programming in object oriented modeling with OWL

2.2.1 OWL Reasoning in SWCLOS

Anonymous Restriction Classes for properties: In CLOS a subClass inherits all the roles of its super class so it's natural that an anonymous restriction class that is individual of owl:restriction on top of CLOS class, exists.

Substantial and non substantial properties: rdfs:subClassOf, owl:intersectionOf, owl:unionOf relations leads to a structural variation in the class – subclass relation and other properties result in reasoning and no variation. So we define the substantial and persistent subsumption by the first ones and non substantial by others [2].

Satisfiability check: Proactive entailment reduces the degree of satisfiability check.

OWL reasoning rules: SameAs relation is reflexive and transitive. So all individuals create a group of sameAs and the information of group is placed in each member. It is the same for EquivalentClass, EquivalentProperty. DifferentFrom and DisjointWith relations are represented without group information because of non transitivity.

2.3 A Reasoning Algorithm for pD*

The Sesame algorithm is a java framework for querying RDF and RDFS data.

2.3.1 Sesame Algorithm

This algorithm uses the dependencies between rules to eliminate certain redundant inferring steps. The dependency is the triggering relation between two rules.

2.3.2 Reasoning Algorithm

After using the optimization discussed above, a forward chaining algorithm in [3] is proposed that is applied on a RDF graph G that contains certain triples and axiomatic triples to finally obtain the pD* closure of this graph.

2.4 Approximation in Reasoning

2.4.1 Approximations Related to the Reasoning Method

Anytime algorithms: anytime algorithms are some sort of algorithms trading execution time for quality of results [4].

Approximate Entailment: can be used to approximate any logical inference problem using the logical entailment operator, including [4]:

- Boolean Constraints Propagation (BCP): a variant of unit resolution.
- S-1- and S-3-entailment
- *Abstraction*: two types of abstraction consist of [4]:
- Syntactic abstraction: syntactic mapping of a problem representation into a simpler one.
- Semantic abstraction: mapping of base language interpretations into abstract language interpretations.

In [5], the authors carried out experiments with approximate deduction techniques on the problem of classifying new concept expressions into an existing OWL ontology using existing ontologies on the web. The method generates two sequences of approximations, one sequence containing weaker concepts and one sequence containing stronger concepts. The method implementation specified that the method is often not suited for Semantic Web reasoning. The goal is to find an approximation strategy that takes the specifics of ontologies into account.

2.4.2 Approximations Related to the Knowledge Base

By reducing the inferring complexity from a knowledge base, the performance reasoning can be enhanced. An area dealing with this problem for knowledge bases written in some logical language is *knowledge compilation*. The goal of knowledge compilation is to translate the knowledge base into (or approximate by) another knowledge base with better computational properties. Knowledge compilation methods consist of [4]:

- Exact knowledge compilation: the original theory is compiled into a logically equivalent theory.
- Approximate knowledge compilation: The underlying idea is that answers to a query can be approximated from two sides: soundness and completeness.

2.4.3 Approximations Related to Language

The idea is based on the well-known trade-off between the expressiveness and the reasoning complexity of a logical language. For example, the logic underlying OWL Full is intractable, so reasoners can use a slightly weaker logic (e.g., OWL Lite). In [6], authors obtained substantially improved reasoning performance by disregarding non-Horn features of OWL DL. An implementation, called Screech being a part of the 'KAON2' OWL tools, was also presented.

3 Our Contribution

3.1 Definitions

Definition 1: TBox is defined in the following format:

$$\Gamma = \langle C, R \rangle \tag{1}$$

C and *R* are the set of axioms concerning concepts and relations respectively.

Definition 2: The axioms in *C* are presented in the following formats:

$$\begin{array}{l} C \equiv Decription\ \ Logic\ \ well-formed\ \ Expression \\ C \sqsubseteq D \\ C \equiv D \end{array} \quad (2)$$

C and *D* are concepts.

Definition 3: The axioms in *R* are presented in the following formats:

$$\begin{array}{l} R \equiv (C, D) \\ R \sqsubseteq S \\ R \equiv S \end{array} \quad (3)$$

C and *D* are concepts; *R* and *S* are relations. If the left side of an axiom is a new name, the axiom is a definition; only one definition should be presented for a concept or a relation.

Definition 4: The world description or *ABox* (*A*) is presented as a set of following axioms:

$$\begin{array}{l} a:C \\ (a,b):R \end{array} \quad (4)$$

a and *b* are instances of concepts; *C* is a concept; and, *R* is a relation in *TBox*.

Definition 5: A conjunctive query *Q* on a *TBox* Γ and an *ABox A* is an expression:

$$Q \leftarrow q_1 \wedge \ldots \wedge q_m \quad (5)$$

Each q_i is represented as *x:C* or *(x,y):R* where *x* and *y* are variables; *C* is a concept; and *R* is a relation in Γ. To avoid the necessity of a new concept definition for each desired condition, q_i can also be presented as *x: (Description Logic well-formed expression)*. It is also possible to use an instance name instead of a variable as a constant value. As an example of conjunctive query, the following one can be considered:

$$Q(X) \leftarrow (X, Y) : father - of \wedge Y : parent \wedge (Y, Maryam) : mother - of$$

Definition 6: The answer set for a conjunctive query includes some tuples. If the variables in the conjunctive query are replaced with the values of the tuples' elements, an expression $_{Q'}$ is resulted for which we have $\Gamma \wedge A \models Q'$.

The origin of above definitions is from [7]; however, we make our required modifications.

3.2 Conjunctive Query Approximation and Its Related Work

To increase the answering performance of the query presented on Γ and *A*, there is an effort to approximate the query more effectively. As far as there is enough resource, the approximation gradually will become more specific. One of the approaches for conjunctive query approximation is the selection of one q_i as primary approximation; in each of the following steps, one of the remaining q_i 's is being conjunct with the previous approximation. In each step, the selection of a q_i which can provide more specific answers with less cost, is desired. So far, the strategies introduced for q_i

selection in successive steps consider only the structure of the query and do not consider the information in Γ and A or the user preferences to provide more specific and favorite answers. In [7], such strategies have been presented. In the paper, the dependencies between variables introduced in the query, are presented as a query graph. The query graph for the example of definition 5 is shown in Fig. 1.

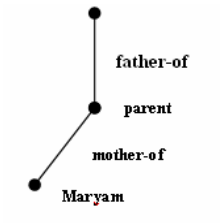


Fig. 1. Query Graph

The root is the goal variable. The introduced strategies are the various modes of the query tree traversal (in the paper, the query graph is supposed to be a query tree by forcing some limitations); including:

- Node expansion approximation
- Arc based approximation:
- Breadth first approximation
- Depth first approximation

For execution, the conjunctive query is transformed into its equivalent concept expression. Then, the expression is classified in the concepts subsumption hierarchy of Γ and corresponding instances are retrieved.

In [8], at first, the conjunctive query is transformed into an equivalent concept expression. Then, some parts of the expression are replaced with $\top$ using a method similar to one proposed in [5]. For the selection of one expression from a group having the same depth to be replaced with $\top$, the expression having less complexity is selected.

3.3 A Heuristic Approach Based on Domain Knowledge and User Preferences for Semantic Web Query Approximation

In our approach, from domain knowledge we mean the information in *TBox* and *ABox* including concepts' and relations' definitions, concepts and relations subsumption hierarchies, and, types of instances or their membership in relations. In each step, the measure for the selection of a q_i to become conjunct with the last step's approximation is its ability to limit the number of possible answers. It is expected that at the end of each step, the answer set would be the most precise answer set. Similar to assumptions in [7], we also assume that the query graph is a tree by considering some limitations; for example, we assume that the root is equivalent to the answer variable and each constant value appears in the query just one time. Also, it is assumed that the structure of the query tree is available during the whole process. The first traversed node is always the root. The selection of a node for being traversed is equivalent to the selection of a component (s) for being conjunct with the previous step's

approximation. As an example, considering Fig. 1, the selection of the root's child node is equivalent to the conjunction of two expressions (X,Y): $father-of$ and Y : $parent$ successively with the previous approximation. If a node is chosen for being traversed, all of the nodes being in the path from the root to it should have been traversed before so that the dependencies between variables would be kept and incorrect deletion of a variable would be avoided; for example, considering Fig. 1, the leaf node should not be traversed before the middle node because variable *y* has not been introduced yet. We define a set of nodes called *Closed* including nodes which are ready for being traversed at the end of each step. At each step, to choose a q_i(s), a score composed of the following factors is computed for each member of *Closed* and the member (s) with highest score is chosen. The factors involved in the final score consist of:

The constant existence in the subtree with the candidate node as its root (Constant): If the subtree with the candidate node as its root has a leaf node with constant label, this factor's value for the node is equal to the ratio of the number of constant leafs in the subtree to the number of constant leafs in the query tree; the more the number of constant leafs in the subtree of a node is, the more this factor's value for the node is. The reason is that in comparison to variables which can be assigned several values, constants decrease the number of possible answers.

$$Cons\tan t = \frac{N_{st}\ (\text{the number of sub - tree constant leafs})}{N_t\ (\text{the number of query tree constants leafs})} \tag{6}$$

The node position in the TBox concepts' subsumption hierarchy (Con_Position): If a node does not have a label specifying its type, the factor's value for the node is zero; otherwise, it is equal to the ratio of the node's depth in the hierarchy (the minimum number of edges between the root (s) and the node) to the hierarchy height (the minimum number of edges between the root (s) and the leafs); the more the depth of the node's label which is a concept in the hierarchy is, the more specific the concept is and accordingly, the number of possible answers are decreased more. As an example, considering Fig. 2: the mother concept is more specific than the woman concept and the number of mothers is more than the number of women in the real world.

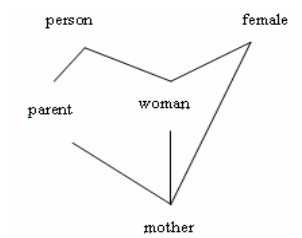


Fig. 2. Concepts' subsumption hierarchy

$$Con\_Position = \frac{D_n\ (\text{the node depth in taxonomy})}{H_t\ (\text{the taxonomy height})} \tag{7}$$

If the node label is a well-formed description logic expression instead of a predefined concept in *TBox*, at first, the expression should be classified in the subsumption

hierarchy and then the factor's value will be computed for the node. (to increase the classification's process performance, the approximation presented in [5] can be used).

The node label definition in TBox (Con_Definition): If a node does not have a label specifying its type, this factor's value for the node is zero; otherwise, it is equal to the number of $\sqcap$ s (*and* operator) in the concept's definition (it is assumed that the concepts' definitions are transformed into their normal form[1] and after the concepts in the concept's definition are replaced with their definitions, the number of $\sqcap$ s is enumerated); the more the number of $\sqcap$ s in a concept's definition is, the higher the value for the node is; because the concept is more specific and the number of possible instances as the answer is decreased more. As an example, when we consider the following two concepts' definitions, the second one is more specific:

$$\text{Man} \equiv \text{Human} \sqcap \text{Male}$$

$$\text{Happy-Father} \equiv \text{Man} \sqcap \forall \textit{has–child.Female} \sqcap \forall \textit{has} - \textit{child} . \textit{Doctor}$$

$$\begin{aligned} &\textit{Con\_Definition} = N_{\wedge} \\ &(\textit{the}\ \text{number of}\ \wedge \text{s in normal form of concept's definition}) \end{aligned} \tag{8}$$

If the node label is a well-formed description logic expression instead of a predefined concept in *TBox*, after replacing the concepts in the expression with their definitions, the expression is transformed into its normal form; then the value is computed.

The precision of components in the definition of node's label (Con_Precision): If a node does not have a label specifying its type, the factor's value for the node is zero; otherwise, it is equal to the ratio of concept's conjunctive components' final scores' sum to the number of the components (concepts' definitions are in normal form); the more the score of conjunctive components in a concept's definition is, the higher the factor's value for the concept is.

$$\textit{Con\_Precision} = \frac{\sum_{i=1}^{\textit{con\_length}} \textit{conjunct}_i \textit{\_score}}{\textit{con\_length}} \tag{9}$$

con_length is the number of conjunctive components in the concept's definition. If the label is a well-formed description logic expression instead of a predefined concept, after transforming the expression to its normal form, the formula is computed.

The edge label position in the TBox relations' subsumption hierarchy (Role_Position): In this factor, the label of edge between traversed nodes in previous steps and the *Closed* set's members is considered. From now, by saying relation we mean the edge label between the specified nodes. This factor' value for the node is equal to the ratio of relation's depth in the relations' subsumption hierarchy (the minimum number of edges between the root (s) and the node) to the relations' hierarchy height (the minimum number of edges between the root (s) and the leafs); the more the depth of the edge's label in the relations' hierarchy is, the more specific the relation is and the number of possible answers are decreased more. As an example, as shown in Fig. 3: the *has-childasfirend* relation is more specific than the *has-child* relation and in real world the number of parents and their children being each other friends is less than the number of parents and their children without considering any condition.

[1] For more information about logical expressions' normal form, refer to the second chapter of [9].

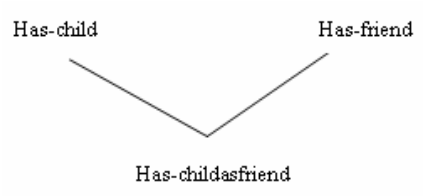


Fig. 3. Relations' subsumption hierarchy

$$Role\_Position = \frac{D_r \text{ (the role depth)}}{H_t \text{ (the taxonomy height)}} \tag{10}$$

The edge label's definition in TBox (Role_Definition): In this factor, the label of edge between traversed nodes in previous steps and the *Closed* set's members is considered. The factor's value for the node is equal to the quotient of two concepts, in the relation's definition (in the form of $R \equiv (C, D)$ or $R \sqsubseteq (C,D)$), final scores' sum by 2; the higher the score of the concepts in the relation's definition is, the higher the factor's value for the node is.

$$Role\_Definition = \frac{(\text{con}_1\_score) + (\text{con}_2\_score)}{2} \tag{11}$$

User preferences: user preferences can be considered in two ways:

- In addition to strict conditions presented in the query, the user can express some characteristics as his preferences for the answers.
- The user specifies the importance of a q_i for him by assigning a weight between 0 and 1 to it.

The application of the first case is in the selection of some answers being more related to the user preferences from the whole correct answer set and also in the answers' ranking according to the user preferences. In our approach, because the whole answer set is returned and the ranking of answers is not in consideration, this type of preference specification does not have application; instead the second case is considered. To keep the original structure of the conjunctive query explained in definition 5, the user should express his preferences about conjunctive components of the query separately. Along with conjunctive query, user presents a weight vector constituting a weight between 0 and 1 for each q_i; for example, consider the following one:

The query is: $Q(X) \leftarrow (X,Y) : father - of \wedge Y : parent \wedge (Y, Maryam) : mother - of$

And, the weight vector is: $W(x) \leftarrow w_{q1} \wedge w_{q2} \wedge w_{q3} \;\; 0 \le w_{qi} \le 1$

The weight expressed by the user can also be considered as a factor for computing the final score of a node (if the node is the identifier of a variable which has a concept as its label, the node weight is presented as the average of node and edge weights).

After computing the mentioned factors for the nodes which are the *Closed* set's members, the final score of a node is computed using a weighted sum formula:

$$node\_score = \frac{\begin{array}{l} w_1 \times Constant + w_2 \times Con\_Position \\ + w_3 \times Con\_Definition + w_4 \times Con\_Precision \\ + w_5 \times Role\_Position + w_6 \times Role\_Definition \\ + w_7 \times user\_assigned\_weight \end{array}}{\sum_{i=1}^{7} w_i} \quad (12)$$

The selection of w_is is very important in the node's final score's value and a considerable attention should be applied in their selection. After computing the final score for the members of the *Closed* set using the above formula, the node (s) having the highest score are selected for being traversed in the next step.

3.4 Guidelines for Query Execution

The queries can be executed using the approach presented in [7]. In addition, in each step, to keep the performance of selecting a conjunctive component to be added to the previous approximation, many of the mentioned factors can be computed offline. Because *TBox* and *ABox* are available before the expression of the query, some factors such as *Con_Position*, *Con_Definition*, and *Con_Precision* for the concepts predefined in *TBox* and *Role_Position* and *Role_Definition* for the relations predefined in *TBox* can be computed offline. Some factors such as *Constant* and factors related to conditions which are description logic well-formed expressions should be computed online at run time necessarily. The computation of *Constant* factor is not so costly, but the computation of factors such as *Con_Position* for conditions presented in the query (because of the necessary pre-classification) may be complex enough to reduce performance which is our goal of approximation. So, the user may become limited in using such conditions; for example, the maximum number of conjunctive components can be used in a condition will be limited to 2.

4 Conclusion and Future Works

In this paper, certain researches that have been done in the field of reasoning in semantic web were verified. We discussed conjunctive queries in semantic web and their approximation methods that help to increase the performance of the reasoning process. So far the methods presented for approximating conjunctive queries use the structure of the query for this purpose. In this paper by using the domain knowledge and user priorities, some heuristics for the better approximation of a conjunctive query were proposed which can be used with syntactic approximations or without them. As future work, the heuristics' implementation in the real world's applications, is considerable.

References

1. Vaneková, V., Bella, J., Gurský, P., Horváth, T.: Fuzzy RDF in the Semantic Web: Deduction and Induction. In: Workshop on Data Analysis (WDA 2005), pp. 16–29 (2005)
2. Koide, S., Takeda, H.: OWL-Full Reasoning from an Object Oriented Perspective. In: 1st Asian Semantic Web Conference. Springer, Beijing (2006)

3. Li, H., Wang, Y., Qu, Y., Pan, Z.: A Reasoning Algorithm for pD*. In: 1st Asian Semantic Web Conference. Springer, Beijing (2006)
4. Wache, H., Serafini, L., Garcia-Castro, R.: Survey of Scalability Techniques for Reasoning with Ontologies. Deliverable of EU-Project KNOWLEDGEWEB (2004)
5. Groot, P., Stuckenschmidt, H., Wache, H.: Approximating Description Logic Classification for Semantic Web Reasoning. In: Gómez-Pérez, A., Euzenat, J. (eds.) ESWC 2005. LNCS, vol. 3532, pp. 318–332. Springer, Heidelberg (2005)
6. Hitzler, P., Vrandečić, D.: The Screech system for approximate reasoning with OWL DL. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 383–397. Springer, Heidelberg (2005)
7. Stuckenschmidt, H., van Harmelen, F.: Approximating Terminological Queries. In: 5th International Conference on Flexible Query Answering Systems, pp. 329–343 (2002)
8. Wache, H., Groot, P., Stuckenschmidt, H.: Scalable Instance Retrieval for the Semantic Web by Approximation. In: Dean, M., Guo, Y., Jun, W., Kaschek, R., Krishnaswamy, S., Pan, Z., Sheng, Q.Z. (eds.) WISE 2005 Workshops. LNCS, vol. 3807, pp. 245–254. Springer, Heidelberg (2005)
9. Baader, F., Nutt, W.: The description logic handbook: theory, implementation, and applications. Cambridge University Press, Cambridge (2003)
10. Horrocks, I., Sattler, U.: Logical Foundations for the Semantic Web-Reasoning Services and Algorithms. University of Manchester (2003)

Semantic Web Services for Handling Data Heterogeneity in an E-Business Framework

Seyyed Ali Rokni Dezfouli, Jafar Habibi, and Soheil Hassas Yeganeh

Sharif University of Technology, Tehran, Iran
rokni@ce.sharif.edu, jhabibi@sharif.edu, yeganeh@mehr.sharif.edu

Abstract. E-business requires interoperability of information systems and, therefore, standardization of information sharing. Several XML-based e-business frameworks are developed to define standards for information sharing within and between companies. These frameworks only standardize structure of messages and aren't able to define semantics. The use of Semantic Web Service (SWS) technologies has been suggested to enable more dynamic B2B integration of heterogeneous systems and partners. We present a semantic B2B mediator based on the WSMX –a SWS execution environment, to tackle heterogeneities in RosettaNet messages. We develop a rich RosettaNet ontology and use the axiomatized knowledge and rules to resolve data heterogeneities. We modeled natural language constraints with formal language that used with WSMX. This is illustrated through a purchasing and shipping scenario. We used a reasoner in interaction with WSMX for automatic consistency checking of instances in run-time. The benefits of applying SWS technologies include more flexibility in accepting heterogeneity in B2B integrations.

1 Introduction

To integrate heterogeneous enterprise information systems, several e-business frameworks have been developed [11, 13]. The e-business frameworks address Business-to-Business (B2B) integration on business process, message and communication levels [13]. Companies have invested considerable amounts of money and resources to implement the B2B integrations based on these e-business frameworks. Due to the flexibility of these e-business frameworks regarding message details and message ordering, considerable effort is required to ensure that the B2B integration details of two partners, match. Traditional B2B integrations suffer from long implementation times and high costs [1], leading to long-term rigid partnerships [2]. This can lead to the situation that buyers use only one supplier as the information systems do not easily support more competitive arrangements.

RosettaNet[1] is a famous and widely used XML-based e-business framework that establishes a common language and standard processes for B2B transactions. It developed the Partner Interface Process (PIP) specification for public business processes between trading partners. The PIP standardizes public business process automation by standardizing business documents, the sequence of sending these documents, and the

[1] www.rosettanet.org

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 453–460, 2008.

physical attributes of the messages that define the quality of service. The content of these PIPs are structurally validated by either DTD for the older and rarely used PIPs or XML Schema for the recently updated and usually used ones[3].

However, the interoperability problems are only partially solved. DTDs and XML Schemas lack expressive power to capture all necessary constraints and do not make all document semantics explicit. Being able to express constraints in machine interpretable format is expected by RosettaNet experts [4].

Semantic technologies and Semantic Web Services (SWS) have been proposed to achieve more dynamic partnerships [15]. The SWS approach based on, for example, OWL-S [5] or the Web Service Modeling Ontology (WSMO) [6] enables annotation of the B2B integration interfaces with semantic annotations. This allows automated or semi-automated mediation. In addition, SWS enables powerful discovery, composition, and selection capabilities of services.

The SWS solution proposed in this paper is based on the Web Service Modelling eXecution environment (WSMX) [14]. WSMX is a reference implementation of the Web Service Modeling Ontology (WSMO) and operates using the Web Service Modeling Language (WSML). WSMO is a meta-model for semantic web services related aspects. We use WSMO meta model because WSMO, WSML and WSMX provide a coherent framework that covers all aspects of semantic web services [4]. We proposed an architecture for using semantic web service technologies in handling data heterogeneity in the RosettaNet framework.

The rest of paper is structured as follows: first we present related work to solving this problem in section 2. Then motivate our solution using an example RosettaNet order and shipping business process in section 3. Section 4 presents the architecture of our semantic B2B mediator. Section 5 introduces steps of data mediation for handling heterogeneous partners. In section 6 we discuss how our solution can be generalized and conclude our paper.

2 Related Works

There are a number of papers discussing the use of semantic web service to enhance current B2B standards. Some concentrate on ontologically annotated B2B standards [8, 10]. Foxvog and Bussler describe how EDI X12 can be presented using WSML, OWL and CycL ontology languages [10]. The paper focuses on the issues encountered when building general purpose B2B ontology, but does not provide an architecture or an implementation. Anicic et al. [9] present how two XML Schema-based automotive B2B standards are lifted using XSLT to OWL-based ontology. They use a two-phase setup and run-time approach similar to ours. The proposed approach in their works is based on different B2B standards and focus only on the lifting and lowering to the ontology level.

Others apply semantic technologies to B2B integrations [7, 8]. Preist et al. [8] presented a solution covering all phases of a B2B integration life-cycle. The paper addresses the lifting and lowering of RosettaNet XML messages to ontologies, but no richer knowledge is formalized or used on the ontological level. Trastour et al. [7, 8] augment RosettaNet PIPs with partner-specific DAML+OIL constraints and use agent technologies to automatically propose modifications if partners use messages differently. Their approach of accepting RosettaNet in its current form and lifting to semantic languages is

similar to ours, but we go further by axiomatizing implicit knowledge and by providing mappings to resolve heterogeneity of data.

3 Motivating Scenario

Consider an organization "A" needs specific components that can be delivered by two suppliers. In current scenario, organization "A" communicates only with one provider that has a pre-agreement with it. In the current situation, the B2B integration only covers purchasing activities and there is no competition for purchasing per delivery basis.

Considering the integrations, the following heterogeneous levels may exist with partners according to general B2B integration levels: *Communication level* interoperation is needed to understand different languages used to describe the messages exchanged and how the message exchange happens. At *Message level* interoperation partners can understand exchanged messages (sometimes referred as business documents or payload). RosettaNet PIPs that define standard inter-company process choreographies and the related schemas for the XML messages exchanged. In this scenario partners use PIP3A4 for Purchase Order Request and PIP3B2 for Shipping Notification. At last, *Business Process level* interoperation is the ability of companies to exchange messages in the right sequence and timing. Partners comply with PIP 3A4 and PIP3B2 standard choreographies.

In presented scenario, all partner uses RosettaNet framework. It is assumed that all use RNIF 2.0 and have no problem in communication level we have no integration problem. Current scenario has some problem in message and business process level of integration of heterogeneous partners. The way that they use same PIPs can be different, as back-end systems of partners are different. In this paper we focus on solving integration problem in message level.

4 Architecture of Semantic B2B Mediator

As you can see in Fig.1 Semantic B2B mediator has three components:

Adapters. Since WSMX internally operates on the semantic level (WSML), adapters provide transformation functionality for every non-WSML message sent to the B2B mediator. Adapters facilitate lifting and lowering operations to transform between XML instances and WSML instances. Furthermore, back-end adapters are necessary to connect the B2B gateway to the back-end applications of the requester.

WSMX. WSMX is the integration platform which facilitates the integration process between different systems. The integration process is defined by the WSMX execution semantics, i.e. interactions of middleware services including discovery, mediation, invocation, choreography, repository, etc.

Reasoner. The reasoner is required to perform query answering operations on the knowledge base, including the collaboration instance data during execution. We use reasoner for operating consistency checking of instances in run-time. The reasoner has to handle WSML**.** As yet simple reasoner for some WSML variant is developed. The latest version of Web Service Modeling Toolkit (WSMT) has included these reasoners. Consistency checking of instances is performed by reasoner.

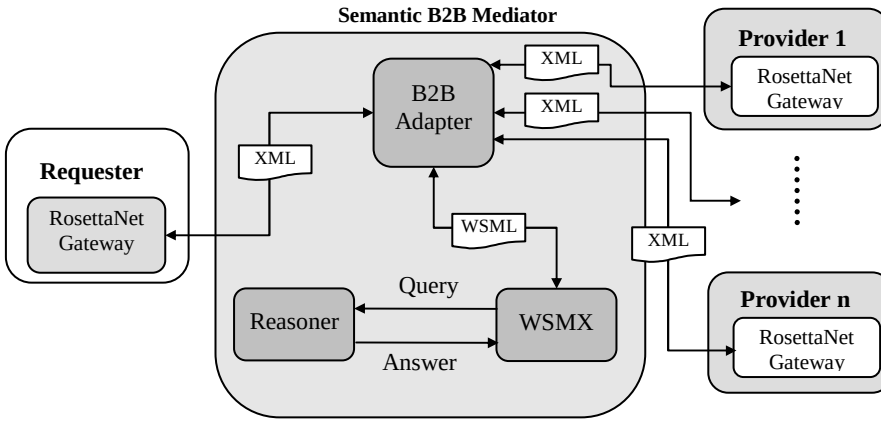


Fig. 1. Semantic B2B Mediator Architecture

There are two main phases to mediate requester and provider partners: mediation setup phase and mediation run-time phase. Setup phase includes defining or extending RosettaNet ontology for specific PIPs with modeling natural language constraints with WSML formal language and defining mapping rules of instances to and from the ontology. In run-time, mapping of instances from and to WSML is preformed, and then consistency checking of instances is performed by reasoner.

5 Data Mediation of RosettaNet Messages

Data mediation means handling heterogeneity of data of various partners (e.g. different currencies) with a mediator. Based on our requirements, domain ontology to formally capture the message content exchanged in any of our RosettaNet collaborations is required. Ideally, existing domain ontologies should be reused.

Creating these domain ontologies requires an expert who first understands specific e-business scenarios and second has knowledge about ontology languages to be able to capture information in messages semantically. However, as yet there is not industry wide recognized ontology for business messages. In sequence, we use ontology that partially defined in [4] and extend it for some constraints and other PIPs. Naturally, this ontology is based on the RosettaNet specification and guideline documents.

5.1 Creating RosettaNet Ontology

The ontology in our scenario ultimately represents simply a different serialization of the information in the RosettaNet framework. This serialization has advantages that it explicitly states logical relationships between elements which can not be expressed in the RosettaNet PIP XML Schemas and DTDs.

For creating this ontology, two mapping is required. First, simple mapping of elements and relations from XML Schema to WSML and second, modeling natural language constraints that XML Schema can not express them.

Translating XML Schema to WSML. Although simple mapping from XML to WSML is available[2], RosettaNet PIPs uses many of associated XML Schemas. For example PIP3A4 have 92 schemas with many namespaces. In sequence we implement a program that explores all files and folders from specific path and aggregates all concepts and relations from ".xsd" files. Three main rules use in this program: First, an XML element is mapped to a WSML concept. Second, an XML child element is mapped to a WSML attribute of the concept corresponding to the XML parent element. It has an inferring type of the WSML concept corresponding to the XML child element. And Third an XML attribute is mapped to a WSML attribute of the concept corresponding to the XML parent element, with a string constraining type.

Modeling constraints. However, the task doesn't end with only simply translating the DTDs or XML Schemas to another richer and formal language like WSML. For complete lifting of XML to WSML we have to model all the constraints on the semantics of the business documents. We include constraints which are not expressible in DTD or XML Schema and included in a natural language in the RosettaNet message guidelines respective PIP. RosettaNet message document for PIP 3A4 contains two natural language constraints named "NetPayment" and "SpecialHandling" constraints. We consider "NetPayment" constraint in this paper, but other constraints can handled in similar way.

NetPayment Constraint. *At least one occurrence of dp:FinancingTerms/dp:Payment Terms/dp:NetTerms/ dp:Day" and "dp:FinancingTerms/dp:PaymentTerms/ dp:Net Terms/ dp:Days" is required.*

This constraint is considered in two optional items that one of them is mandatory. This constraint can solve in different form in WSML variants. As you see in Table 1, we can express these constraints in WSML-Full in several ways.

"?x" is any variable instance of concept NetTerm. "?x[Day hasValue ?y]" means, Day attribute of instance "?x" has a value called "?y". In WSML syntax[3] use this format for expressing the existence of an attribute of an instance.

These axioms can't express in WSML-Rule in sequence of three syntax limitation:

- Head formula must not contain a Universal Quantification.
- Head formula must not contain a disjunction.
- Body formula must not contain a negation.

On the other hand we want to evaluate our solution and must use reasoner for consistency checking. WSML-Full reasoner has not been developed, yet. Thus, we have to define constraints in WSML-Rule. We define a default value that is assigned to attributes of concepts when they do not have a value. This default value is a pattern that isn't use practically. For example Days must be a positive integer and XML Schema of it, checks this limitation. We set default value of Days to -1. Setting of default values performed in adapter and these pre-set values only use in WSML form. With this lemma, we express this constraint in Table 1 by WSML-Rule. Modeled constraints can verify by reasoner.

[2] `http://purl.oclc.org/net/wsmo/serializers/xml-to-wsml.xsl`
[3] `http://www.wsmo.org/wsml/wsml-syntax`

Table 1. NetPayment constraint in WSML-variants

```
concept NetTerm
   Day ofType (0 1) _date
   Days ofType (0 1) _integer
axiom ValidNetTerm
definedBy
   forall ?x (?x memberOf NetTerm
      implies
         ?x[Day hasValue ?y]  or ?x[Days hasValue ?y]).
axiom NewValidTerm2
definedBy
    !- ?x memberOf  NetTerm and
      neg (?x[Days hasValue ?y] or ?x[Day hasValue ?y]).
axiom ValidNetTerm_WSML-Rule
definedBy
    !- ?x memberOf NetTerm  and
    ?x[Days hasValue -1] and
     ?x[Day hasValue _date("0-0-0")].
```

5.2 Mapping Rules and Consistency Checking

The mapping rules need to be defined for the run-time phase to lift RosettaNet instance messages to WSML ontology and lower it back to the XML level respectively. We perform this by using XSLT stylesheets. Table 2 shows such an example this mapping.

Table 2. XML-Schema PIP mapping extract

```
<xsl:for-each select="dp:FinancingTerms/dp:PaymentTerms/dp:NetTerms">
    instance NT_instance memberOf NetTerm Day hasValue
   <xsl:value-of select="dp:Day"></xsl:value-of>
       Days hasValue <xsl:value-of select="dp:Days"></xsl:value-of>
</xsl:for-each>
```

In the lowering of messages, as the information does not lose, mapping rule is simple. The mapping rules need to be registered in the WSMX ontology repository for run-time mappings. As the product information definitions in all PIPs are similar, these mapping templates can be reused with all the PIPs. With small modification it is easy to create templates for other e-business frameworks as well.

We need a method to check instances are compliant with business constraints. For this purpose we use WSML-Rule reasoner for explore instances and find ones that violate some constraints. We generate one query for finding all instances form all concepts and request reasoner to answer us with this instances. If there is any instance that violates any axioms, the reasoner can't answer us and generate an error message. By means of this query we can know whether consistency problems are existing or no.

6 Conclusion and Discussions

The scenario discussed in the paper highlights the problems currently observed in RosettaNet collaborations. This problem leads to decrease the chance of competitive sellers that have not any pre-agreement with buyer. Using semantic web service technologies helps sellers to being able to dynamic integrate to buyers without having to make potentially costly changes to their current integration interfaces.

We have used one specific RosettaNet PIP for our solution. As the structure of PIPs is very similar, the results presented in this paper are applicable to the entire RosettaNet framework. By using a formal language such as WSML we can express implicit constraints in a formal and with no ambiguity.

Our semantic B2B mediator allows partners to tackle heterogeneities in RosettaNet interaction messages. The solution relies upon a formalized RosettaNet ontology including axiomatized knowledge and rules to resolve data heterogeneities. We showed how to model and formalize constraints are expressed in natural language. We use a B2B adapter that for transformation of instances from XML to WSML and vice versa. By using a reasoner consistency checking of instances in run-time is carried out.

For future work, we want to classify constraints in RosettaNet PIPs and suggest an automatic method for modeling of constraints base of their class. As the integration is performed in three levels, for future work we can to extend this mediator with process mediation. In practice, several e-business frameworks maybe exist. We have to only define B2B adapter for any framework. Another work is to use other e-business frameworks.

Acknowledgement

This material is based on works that supported by Iran Telecommunication Research Center.

References

1. Preist, C., Cuadrado, J.E., Battle, S., Williams, S., Grimm, S.: Automated business-to-business integration of a logistics supply chain using semantic web services technology. In: Borrione, D., Paul, W. (eds.) CHARME 2005. LNCS, vol. 3725, pp. 987–1001. Springer, Heidelberg (2005)
2. Kotinurmi, P., Vitvar, T., Haller, A., Boran, R., Richardson: Semantic web services enabled b2b integration. In: Lee, J., Shim, J., Lee, S.-g., Bussler, C.J., Shim, S. (eds.) DEECS 2006. LNCS, vol. 4055, pp. 209–223. Springer, Heidelberg (2006)
3. Damodaran, S.: B2B integration over the Internet with XML: RosettaNet successes and challenges. In: 13th International World Wide Web Conference – Alternate track papers & posters, pp. 188–195. ACM Press, New York (2004)
4. Haller, A., Kotinurmi, P., Vitvar, T., Oren, E.: Handling heterogeneity in RosettaNet messages. In: 2007 ACM Symposium on Applied Computing, pp. 1368–1374. ACM Press, New York (2007)

5. Martin, D., et al.: Owl-s: Semantic Markup for Web Services. Member submission. W3C (2004), http://www.w3.org/Submission/OWL-S/
6. Roman, D., et al.: Web service modeling ontology. Applied Ontologies 1(1), 77–106 (2005)
7. Trastour, D., Bartolini, C., Preist, C.: Semantic web support for the business-to-business e-commerce pre-contractual lifecycle. Computer Networks 42(5), 661–673 (2003)
8. Trastour, D., Preist, C., Coleman, D.: Using semantic web technology to enhance current business-to-business integration approaches. In: International Enterprise Distributed Object Computing Conference, pp. 222–231 (2003)
9. Anicic, N., Ivezic, N., Jones, A.: An architecture for semantic enterprise application integration standards. In: Interoperability of Enterprise Software and Applications, pp. 25–34. Springer, Heidelberg (2006)
10. Foxvog, D., Bussler, C.: Ontologizing EDI: first steps and initial experience. In: International Workshop on Data Engineering Issues in E-Commerce and Services, pp. 49–58 (2005)
11. Shim, S.S.Y., Pendyala, V.S., Sundaram, M., Gao, J.Z.: Business-to-Business E-Commerce Frameworks. IEEE Computer 33(10), 40–47 (2000)
12. Shariq, O., et al.: Investigating semantic web service execution environments: a comparison between WSMX and OWL-S tools. In: 2nd International Conference on Internet and Web Applications and Services, pp. 31–38 (2007)
13. Medjahed, B., Benatallah, B., Bouguettaya, A., Ngu, A.H.H., Elmagarmid, A., Elmagarmid, K.: Business-to-business interactions: issues and enabling technologies. VLDB Journal 12, 59–85 (2003)
14. Haller Haller, A., Cimpian, E., Mocan, A., Oren, E., Bussler, C.: WSMX – A semantic service-oriented architecture. In: 3rd International Conference on Web Services, pp. 321–328. IEEE Computer Society, Los Alamitos (2005)
15. Vitvar, T., Moran, M., Zaremba, M., Haller, A., Kotinurmi, P.: Semantic SOA to promote integration of heterogeneous B2B services. In: 4th IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services, pp. 451–456 (2007)

Challenges in Using Peer-to-Peer Structures in Order to Design a Large-Scale Web Search Engine*

Hamid Mousavi and Ali Movaghar

CE Department, Sharif University of Technology, Tehran, Iran
h_mousavi@ce.sharif.edu, movaghar@sharif.edu

Abstract. One of the distributed solutions for scaling Web Search Engines (WSEs) may be peer-to-peer (P2P) structures. P2P structures are successfully being used in many systems with lower cost than ordinary distributed solutions. However, the fact that they can also be beneficial for large-scale WSEs is still a controversial subject. In this paper, we introduce challenges in using P2P structures to design a large-scale WSE. Considering different types of P2P systems, we introduce possible P2P models for this purpose. Using some quantitative evaluation, we compare these models from different aspects to find out which one is the best in order to construct a large-scale WSE. Our studies indicate that traditional P2P structures are not good choices in this area and the best model may be the use of a special case of Super-Peer Networks which is yet conditioned on the peers' active and trustful contributions.

Keywords: Web Search Engines; Peer-to-Peer Systems; Super-Peer Networks.

1 Introduction

Dynamicity [1], fast-growing nature [2], and tremendous size [3] of the Web make the use of *Web Search Engines* (WSEs) an obligation for almost all of its users. Today's commercial WSEs are covering less than 1 percent of the existing Web pages [4][5][6][20]. To improve this coverage, some parallel approaches are being used in most of these WSEs. However, central or intra-site parallel techniques can not completely afford the current needs of users [7]. To improve the performance, some distributed techniques have been employed in WSEs too [7][8][9]. In recent years, some studies have also been focused on using *Peer-to-Peer* (P2P) structures in WSEs [4][10][11]. Although there are some works trying to use P2P systems in some parts of WSEs, the fact that P2P structures can be beneficial for WSEs is still a controversial subject.

Currently, there are three main generations of P2P systems: *Pure*, *Hybrid*, and *Super-Peer* networks. A few analyses have been done on the feasibility of using these structures in WSEs. Estimating the size of the problem, Li *et al.* showed that naive implementations of WSEs are not feasible in traditional P2P structures [4]. However, they have considered only one type of P2P systems in their research.

* You can find the complete version of this paper in the first author's website.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 461–468, 2008.

In this paper, we investigate challenges in using existing P2P structures in large-scale WSEs. From a large-scale WSE, we mean the one which works as good as the current commercial WSEs and can scale with a reasonable cost when the number of Web pages and/or the number of users are increased. To this end, after discussing the main characteristics of both P2P structures and large-scale WSEs, we introduce possible P2P-based models for WSEs and discuss their pros and cons. Using more specific analysis and some quantitative evaluations, we try to compare the proposed models with each others and also with traditional central and distributed models. Briefly, the results show that Pure and Hybrid P2P structures are not good choices for large-scale WSEs. However, Super-Peer networks, with relatively strong super peers, seem to be a good alternative if the indices of the Web pages are distributed only over the super peers and peers actively contribute.

The rest of the paper is organized as follows. After providing some backgrounds and and Common problems of P2P-based WSEs in section 2, we discuss the existing constraints and our assumptions in section 3. Comparison metrics for different WSEs are discussed in section 4. We introduce possible P2P-based models in sections 5 and 6. More Comparisons are included in section 7. Finally, we conclude the paper in section 8.

2 Background

Commercial WSEs, such as Google [12], are usually crawler-based. In these WSEs, some Web pages are downloaded with a specific policy. Then, these crawled pages are indexed and their *inverted index* is constructed. For each word, inverted index contains a list of those documents' identifiers (docID) which contain the word. Thus, for a query search, it is enough to compute the intersection of related documents' lists for each term of the query. These results are then ranked based on their importance and relevance.

In a distributed manner, in addition to these steps, we should address the way that the WSE is going to route the query to the responsible nodes and collect the results from them. The other issue is the way in which the inverted index is going to be partitioned over nodes. There are basically two partitioning approaches: *partitioning by documents* and *partitioning by keywords*. In the former case, each node contains its local inverted index of some specific documents which are usually its own gathered pages. In the latter case, each node is responsible for some specific words, and stores the inverted index of these words. Usually, a *Dynamic Hash Table* (DHT) would be used to map words to nodes in this case.

3 Constraints and Assumptions

The number of users and consequently the number of queries to WSEs are also growing rapidly. In 2005, [20] reported that users of the Web are searching 4.5 billion pages every month, that is more than 1730 queries per second. Considering the growth of the users' number, reported by CIA [21], from 1.07 billion in 2005 to 1.35 billion in 2007, this value in 2007 should be more than 2180 queries per second.

It is estimated that the surface web contains more than 37.5 billion pages with average page size of 43 kilobytes [13]. This is actually something around 1612 terabytes of indexable data. Therefore, to cover only the current indexable surface Web, a good WSE should be able to cover at least 37.5 billion pages. According to the estimation of the number of total queries per second, we consider that a large-scale WSE should be able to respond at least 1000 queries per second. Considering each document contains 1850 words [13] and its identifier (*docID*) occupies 20 bytes, we can conclude that the size of the inverted index is around 1387 terabytes ($1850\times20\times37.5\times10^{9}$.) Another important parameter is the per-query communication budget. In [4], 1 megabyte is used as a very optimistic estimation. In [14], few hundred kilobytes is used as a realistic one. We consider this value to be 1 megabyte. The size of each query is also considered to be 100 bytes.

4 Comparison Metrics for Large-Scale WSEs

For a better comparison of the different models for large-scale WSEs, we have to identify important metrics. Table 1 shows some of these metrics. In Addition to these, there are some other minor characteristics too. Politeness is one of them which deals with treating Web servers politely. For example, overloading a Web server is an impolite action [5]. Portability is the other important characteristic. This will be more important for the part that should be spread over a distributed or P2P networks, since different platforms may be involved. Another important feature of good search engines can be the support for different formats of documents in the Web [15].

5 Simple Models

Li *et al.* in [4] showed that naive Web search algorithms are not feasible in Pure P2P systems. In this section, we will repeat their analysis with newer statistical information. We also investigate the feasibility of designing WSEs using Hybrid P2P systems.

5.1 Pure P2P and Partitioning by Document

As the title indicates, in this model, each peer in a Pure P2P structure has the local inverted index of its containing documents. Thus, each query should be broadcasted (flooded) to all the peers. Now, suppose that a query of 100 bytes should be flooded to N_P peers, where N_P is the number of peers in the network. Thus, due to the 1 megabyte per-query communication budget, we can have at most N_P=10,000 peers. Note that we do not even consider the communication overhead of gathering results of the peers and also, such a communication budget is very optimistic for a pure P2P structure considering the ad hoc connections in this structure. Nevertheless, to store the entire inverted index, each peer should have 138 GB of free memory, in average. Obviously, such an assumption makes no sense for ordinary peers which are usually PCs and laptops. Practical examples of this kind of structures also showed that it is not scalable at all [35]. This is confirmed in [4] too. Note that some compression ideas may be applied to the model, but the number of peers remains restricted and the inscalability of the model remains unsolved.

Table 1. Comparison Metrics list for Large-Scale WSEs

Name	Description.
Accuracy	The most important characteristic of a WSE is the quality of the results provided for queries. Informally, accuracy means to provide the best matching results for a query. To reach a good accuracy, a WSE has to have a good coverage, up-to-dated pages, and also a good ranking algorithm. Here, to separate accuracy from coverage and freshness issues, we define it to be the quality of results the WSE provides for queries from the currently collected pages.
Response Time	In WSEs, the response delay for queries should be only a few seconds, considering both communication and processing delays.
Throughput	It can be defined as the number of queries can be handled per second.
Freshness	We define the freshness of a WSE to be the percentage of up-to-dated pages of the currently collected pages. The most important issue dealing with freshness is the revisiting policy. Revisiting policy indicates how frequently the pages should be revisited for change detection.
Coverage	Coverage usually is considered as the ratio of the number of gathered pages to the entire Web's pages number.
Scalability	WSEs should be scalable in two major aspects: number of queries per second and number of indexed pages. Informally, a scalable WSE can be defined as the one which is capable to increase total throughput under an increased load with a reasonable cost.
Cost	Costs are emanated from the resources (Development and maintenance staffs, bandwidth, storage, and CPU cycle) needed to develop and maintain the system.
Fault Toleration	A WSE should be fault tolerant, (at least with no single point of failure.) It should also be robust enough.

5.2 Pure P2P and Partitioning by Keyword

Since the inverted index is distributed over peers based on keywords in a Pure P2P structure, responding to a query containing only one term is straightforward. Using a DHT, the responsible peer for the term is found, and the results will be fetched from the peer. The communication cost of this action is almost optimal. However, this is not usually the case. As reported in [4] and [16], more than 60% of the queries consist of more than one term. Actually, queries contain 2.35 words in average [17]. For queries containing multiple terms, common procedure is to send each term to its related peer, and the smallest list of the results is transferred to the node that includes the second smallest list. Then, the intersection of these two lists will be sent to the node containing the next smallest list, and so on.

5.3 Hybrid P2P and Partitioning by Document

Two cases can be considered: storing the entire inverted index in the central part of the structure or distributing it over the peers. The former case is actually a simple extension for central approach, that is the main operations and also data are still handled in the central part. In the latter case, the central control acts only as a simple coordinator. In this case, the central control can alleviate some of the problems mentioned in part A, but it has some kind of single point of failure as well. It may also aggravate the inscalability at the same time. Depending on the responsibility of the central control, it may have relatively high cost too.

We believe that both cases are not scalable, and sophisticated techniques are needed to make it possible to use them in large-scale WSEs, however the first one has some characteristics which are worthy to be discussed. We should firstly note that the

most important benefit of the Hybrid P2P networks is that the data flow is still distributed but control flow is not, which make it more manageable than Pure P2P. Thus, one may say that this case can not be called P2P at all since the data is not distributed.

In the first case, keyword searching algorithm is the same as in the centralized (or clustered) WSEs. However, in this case, some parts of WSEs can be implemented in a distributed manner. The most probable part is the crawler. In the simplest case, the central part can assign each peer to crawl some part of the Web. Peers, after crawling their portions, return the compressed indexed pages to the central control. Obviously, this will distribute the traffic over the peers and increase the freshness of the WSE. It may also reduce the cost of the WSEs.

Table 2. Comparing different models for large-scale WSEs from different aspects

Model Name	properties	Accuracy	Response delay	Freshness	Coverage	Scalability	Cost	Fault toleration
central	Central or intra-site parallel	Very High	$\propto 5K$	$\propto BW_{SP}/PS$	$\propto M_{SP}$	Middle Low	Very High	Low
Dist	Partitioned by document	High	$\propto 5K+6N_{SP}$	$\propto N_{SP}\times BW_{SP}/PS$	$\propto N_{SP}\times M_{SP}$	High	High	High
Pure	Partitioned by document, Pure P2P network	Low	$\propto 5K+6N_P$	$\propto N_P\times BW_C/PS$	$\propto N_C\times M_C$	Low	Low	Very High
Hybrid-cent	Inverted index is centralized, Hybrid P2P network	Very high	$\propto 5K$	$\propto BW_{SP}/(IPS/CC)$	$\propto M_{SP}$	Middle Low	Very High	Low
Hybrid-dist	Partitioned by document, Hybrid P2P network, Inverted index is distributed	Low	$\propto 5K+6(1+N_C)$	$\propto N_C\times BW_C/PS$	$\propto M_{SP}$	Low	Middle	Very High
SP-client	Partitioned by document, Super-peer network, Inverted index is distributed over clients	Low	$\propto 5K+6(N_{SP}+N_C)$	$\propto N_C\times BW_C/PS + N_{SP}\times BW_{SP}/PS$	$\propto N_{SP}\times M_{SP}+N_C\times M_C$	Low	Middle Low	Very High
SP-SP	Partitioned by document, Super-peer network, Inverted index is distributed over super peers	High	$\propto 5K+6N_{SP}$	$\propto PPS$	$\propto N_{SP}\times M_{SP}$	High	High	High

5.4 Hybrid P2P and Partitioning by Keyword

Since in this case the inverted index is still distributed over ordinary peers, the problems of the DHT-based WSEs in Pure P2P systems remain as the same. In this case, the central control can not do much for intersection problem too. Besides, with respect to the responsibility of the central control, we may have different kinds of single point of failure and cost problems too.

6 Super-Peer-Based Models

In this section, we introduce some possible Super-Peer-based models for large-scale WSEs. Through this section, we suppose that the number of super peers and ordinary peers (clients) are N_{SP} and N_C respectively.

6.1 Distributing over Clients and Partitioning by Document

In this case, the inverted index is distributed over all of the peers based on documents. To answer a query, too many peers are involved. Thus, we have the same problems of Pure P2P models. The most important one is that N_C is restricted because of the high communication overhead of flooding the queries. The model can just solve the single point of failure problem with respect to the Hybrid ones. Note that in this case, some optimizations such as caching may be applied to alleviate the problems, but because of the dynamicity and the size of the Web, they may not be useful enough. Unless, we cache all the inverted index of clients in powerful and permanent super peers which actually means that the inverted index is distributed only over super peers. We study this case in part C.

6.2 Distributing over Clients and Partitioning by Keyword

Obviously, this model has the same problems we mentioned in part B of the previous section. The most important one is the high cost of the intersection operations. Again, some optimizations can be used. However, for the same reasons they should be so sophisticated and probably expensive to solve the mentioned problems.

6.3 Distributing over Super-Peers and Partitioning by Document

The first two models of this section have serious scalability problem because of that too many peers have to deal with answering a query. Thus, distributing the inverted index over a limited number of peers sounds good. In this case, super peers are responsible to store the inverted index and to answer the queries. Obviously, we have to have stronger and more permanent super peers which increase the cost, as a result. Each super peer can be a cluster of workstations. Other peers (clients) can perform other tasks of the WSE such as crawling, indexing, and etc.

Since a query should be flooded to N_{SP} super peers, we have only $100N_{SP}$ bytes of flooding overhead for each query. Even if there exist 1000 super peers, the overhead will be only 100 kilobytes which is affordable. Most of the overhead, in this case, is the overhead of gathering the results. Considering that users are interested to see only the first 10 results in average [18] and each super peer has more than 10 results for a query, in the worst case, we need to transfer $5N_{SP}$ kilobytes of data for each query if each result contains 500 bytes of data. Thus, Considering 1 megabyte per-query communication budget, N_{SP} should be less than 200 which is not irrational.

Using compression and some other techniques can also reduce both flooding and result gathering overheads. The simplest optimization can be that if it is desired to find the best K results from N_{SP} super-peers, it is usually enough to pick up only the first $\lceil K / N_{SP} + 1 \rceil$ results from each super peer, considering the page distribution is uniform in all super peers. It is worthy to note that since the super peers are strong nodes (or clusters,) there is much more hope for them to be connected with more powerful links to each other, so we may have further budget for per-query communication with respect to the traditional P2P systems. Replication can also be used to increase the per-query communication budget. This kind of optimizations may make it possible to have even more than 1000 super peers.

6.4 Distributing over Super-Peers and Partitioning by Keyword

In this model, there is less number of nodes involving in answering a query with respect to the previous DHT-based models. Thus, there is a higher probability for the terms of a multi-term query to have the same responsible peer. A good assignment of the terms to the nodes can also increase this probability. However, it seems that the intersection problem can not be solved in any of these ways. Other important characteristics of the model are similar to the previous one.

7 Comparison and Discussion

So far, it seems that models, which partition the inverted index by keyword over peers, are not feasible, or at least need very sophisticated and expensive approaches. That is, DHT-based approaches may be not appropriate enough to be used in large-scale WSEs. This is primarily because of that most of queries have more than one term and the number of results for each term in the Web is usually high. Thus, computing the intersection of these results would be very expensive. In the rest of this section, we put DHT-based models behind, and continue with other models for detail comparison. We can also see that the number of peers involved in answering queries should be limited. Otherwise, the communication costs will excess from the rational limits if we want to have fast query response and high accuracy. This claim has been approved in [4] and [19].

8 Conclusion and Future Work

In this article, we tried to investigate challenges in using P2P systems in order to design large-scale WSEs. Comparisons imply that *dist* and *SP-SP* are the most feasible models for large-scale WSEs. *SP-SP* can also work a little better than *dist*, if we have the peers active contributions. However, these two models are too close to each other, the only deference is that in *SP-SP*, we have client peers which can help in crawling, indexing, and maybe ranking algorithms. This can improve the freshness of the WSE and reduce its total cost. Besides, *hybrid-cent* may be a good extension for *cent* model. Maintaining all the features of a central approach, it can improve its freshness and decrease the cost of the model.

Table 2 shows the brief sketch of our comparison results. As the results show, the P2P approaches, in which data is distributed over all the peers, seem not to be feasible to be used in WSEs. It is also shown that DHT-based approaches are not good choices for large-scale WSEs. On the other hand, the results imply that *dist* and *SP-SP* models are of the best possible models for large-scale WSEs. Note that in *SP-SP* model, the data is not distributed over all the peers. This is not usually the case in Super-Peer Networks. Thus, the model is still more distributed and less Super-Peer-based.

References

1. Brewington, B.E., Cybenko, G.: How dynamic is the Web? In: Procs of 9th International World-Wide Web Conference (May 2000)
2. Cyveillance. Sizing the internet. White paper (July 2000), http://www.cyveillance.com/
3. Lyman, P., Varian, H.R., Charles, P., Good, N., Jordan, L.L., Pal, J.: How much information? (2003)
4. Li, J., Loo, B.T., Hellerstein, J., Kaashoek, F., Karger, D.R., Morris, R.: On the feasibility of P2P Web indexing and search. In: Procs of the 2nd Int. Workshop on P2P Systems (2003)
5. Ye, S., Lu, G., Li, X.: Workload-aware Web crawling and server workload detection. In: Network Research Workshop, 18th Asian Pacific Advanced Network Meeting (July 2004)
6. Cho, J., Garcia-Molina, H.: The evolution of the Web and implications for an incremental crawler. In: Procs of 26th International Conference on VLDB, Cairo, Egypt, pp. 200–209 (2000)
7. Wu, L.S., Akavipat, R., Menczer, F.: 6S: Distributing crawling and searching across Web peers. Web Technologies, Applications, and Services, pp. 159–164 (2005)
8. Papapetrou, O., Samaras, G.: Distributed location aware Web crawling. WWW (Alternate Track Papers & Posters,) pp. 468–469 (2004)
9. Wang, Y., DeWitt, D.: Computing PageRank in a distributed internet search system. In: Procs of the International Conference on Very Large Databases (August 2004)
10. Suel, T., Mathur, C., Wu, J., Zhang, J., Delis, A., Kharrazi, M., Long, X., Shanmugasunderam, K.: Odissea: A peer-to-peer architecture for scalable Web search and information retrieval. Technical Report, Polytechnic University (2003)
11. Sankaralingam, K., Sethumadhavan, S., Browne, J.C.: Distributed PageRank for p2p systems. In: Procs of the 12th IEEE International Symposium on High Performance Distributed Computing, Seattle, Washington, USA (June 2003)
12. Brin, S., Page, L.: The anatomy of a large-scale hypertextual Web search engine. In: Procs of the 7th World Wide Web Conference, vol. 30(1/7), pp. 107–117 (1998)
13. Mousavi, H., Rafiei, M., Movaghar, A.: Characterizing the Web Using a New Uniform Sam-pling Approach. In: Procs. of Comsware 2007, India (2007)
14. Tang, C., Xu, Z., Mahalingam, M.: pSearch: Information retrieval in structured overlays. In: First Workshop on Hot Topics in Networks (HotNets I), Princeton, NJ (October 2002)
15. Dikaiakos, M., Stassopoulou, A., Papageorgiou, L.: An investigation of Web crawler behavior: characterization and metrics. Computer Communications 28(8), 880–897 (2005)
16. Gulli, A., Signorini, A.: The indexable Web is more than 11.5 billion pages. In: WWW (Special interest tracks and posters), pp. 902–903 (2005)
17. Silverstein, C., Henzinger, M., Marais, H., Moricz, M.: Analysis of a very large Web search engine query log. SIGIR Forum 33(1), 6–12 (1999)
18. Balke, W.T., Nejdl, W., Siberski, W., Thaden, U.: Progressive distributed top-k retrieval in peer-to-peer networks. In: Procs. of 21st Int. Conf. on Data Engineering, Tokyo (2005)
19. Craswell, N., Crimmins, F., Hawking, D., Moffat, A.: Performance and cost tradeoffs in Web search. In: Procs. of the Australasian Database Conference ADC 2004 (2004)
20. The Search Engine Watch Website, http://www.searchenginewatch.com
21. http://cia.gov/cia/publication/factbook

Sparse Sinusoidal Signal Representation for Speech and Music Signals

Pejman Mowlaee, Amirhossein Froghani,
and Abolghasem Sayadiyan

Electrical Engineering Department
Amirkabir University of Technology, Tehran, Iran
pejman_mowlaee@ieee.org, pouyaforghani@yahoo.com,
eea335@cic.aut.ac.ir

Abstract. We present a sparse representation called Fixed Dimension Modified Sinusoid Model (FD-MSM) for parametric analysis of audible signals including speech, music and mixtures. Compared with other analysis models, the proposed scheme is both pitch independent and appropriate for sparse signal representation commonly found as a favorable choice for speech enhancement and sound separation. Using the state-of-the-art Principle Component Analysis (PCA) it is demonstrated that FD-MSM signal representation is equivalent to a non-linear mapping into sinusoidal subspace which preserves those components with largest eigenvalues by projecting the signal components into the corresponding eigen-vectors. Conducting subjective experiments, we observed that the resulting signal is perceptually indistinguishable from the original ones.

Keywords: Sinusoidal subspace, STFT, Principle Component Analysis, Sparse Representation, SSNR.

1 Introduction

One of the promising methods in speech processing is proven to be sinusoid model since many musical instruments produce harmonic or nearly harmonic signals with relatively slowly varying sinusoidal partials. On the other hand, Frequency analysis is, roughly speaking, the process of decomposing a signal into frequency components, that is, complex exponential signals or sinusoidal signals. Hence, in many applications finding a sparse representation for the audio signals is a must since it can easily result in lower computational complexity or selecting better features. As a result, sinusoidal modelling offers a parametric representation of audible signal components such that the original signal can be recovered by synthesis and addition of the components [1-2]. The most prominent classic sinusoidal model includes: (1) McAulay and Quatieri [3] and (2) Smith and George [4].

In model presented by McAulay and Quatieri called *Sinusoidal Transformation System* (STS) [3], all peaks from *Short-time Fourier Transform* (STFT) in the spectrum are marked, then the peaks whose occurrences are close to the pitch values and its harmonics are held while the remaining peaks are discarded. However, for mixed

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 469–476, 2008.

audio signals, it fails to act properly due to its pitch dependency. However, the other sinusoidal model introduced by Smith and George, namely *Analysis-by-Synthesis/Overlap-Add* (ABS/OLA), has proven to be successful [4]. In contrast to STS proposed in [3], it uses a successive approximation-based analysis by synthesis procedure to determine model parameters. However, the computational load for estimating sinusoid parameters remains an obstacle due to exhaustive frequency search. It also requires initial pitch frequency estimation which is susceptible to gross errors for multi-speaker and speech + noise signals [4].

None of the above mentioned sinusoidal approaches are capable to extract fixed number of features for the underlying model. This drawback in turn results in a significant degradation in clustering performance since all model-based speech processing techniques employ statistical modelling techniques. Hence, we have recently proposed a modified version of sinusoidal model called *Fixed Dimension Modified Sinusoid Model* (FD-MSM) in [7], based on model proposed by McAulay and Quatieri [3], including inputs other than speech signals, i.e. music or mixtures. In this paper we study the sparse representation nature of FD-MSM which arrives at a lower dimension while preserving natural signal quality as close as possible.

The paper is organized as follows: The following section summarizes the state-of-the-art sinusoidal models. Section 3 is dedicated to the important concept of sparse representation of audio signals in sinusoidal space. In Section 4, objective and subjective results are reported and Section 5 concludes

2 Sinusoidal Model for Speech

Independent of which approach is used for analysis of speech signals, the spectrum envelope is known as a key feature, generally obtained by method proposed by McAulay [3]. Sinusoidal model represents a sound signal as a set of sinusoids parameterized by amplitude, frequency and phase trajectories carried out over short frames assuming short-time stationarity; under this assumption, frames of speech are modeled as a sum of constant-amplitude and frequency sinusoids [3]. Assuming the analysis frame $\approx$5-40$^{\text{ms}}$, the speech segment of frame k, $\mathbf{s}^{\text{k}}$(n) will be:

$$s^k(n) = \sum_{j=0}^{M} A_j^k \cos(2\pi \mathrm{f_j^k n/f_s} + \varphi_j^k), \qquad n = 0,...,2\times \mathrm{fl} \tag{1}$$

where f_j is the frequency, M is the number of sinusoids, A_j is the spectrum envelope, φ_j the phase sampled at k$^{\text{th}}$ frame, j the index number, and f_s the sampling frequency. The drawback for such representation is its inherent high computational complexity due to storage of 3M values of spectrum parameters sampled at each frame.

3 Sparse Representation for Signals in Sinusoidal Space

If we consider the spectrum of a harmonic process, we note that it consists of a set of impulses with a constant background level at the power of the white noise. As a result, the power spectrum of complex exponentials is commonly referred to as a *line spectrum* and such signal can be expressed as:

$$x(n) = \sum_{l=1}^{L} e^{j2\pi n f_l} + w(n) = s(n) + w(n) \tag{2}$$

where $\mathbf{w}(n)=[w(n)\ w(n+1)\ \cdots\ w(n+L-1)]^T$ is the windowed vector of white noise and

$$\boldsymbol{V}(f_i) = [1, e^{j2\pi f_i} \ldots, e^{j2\pi f_i (N-1)}]^T,\ 1 \le i \le L \tag{3}$$

is the time-window frequency vector. Note that $\mathbf{v}(f_i)$ is simply a length-N DFT vector at frequency f. We differentiate here between $\mathbf{s}(n)$, consisting the sum of complex exponentials, and the noise component $\mathbf{w}(n)$, respectively. The autocorrelation matrix of the model can be written as

$$\boldsymbol{R}_x = \mathrm{E}\{\boldsymbol{x}(n)\boldsymbol{x}^{\boldsymbol{H}}(n)\} = \boldsymbol{R}_S + \boldsymbol{R}_w \tag{4}$$

$$\boldsymbol{R}_x = \sum_{l=1}^{L} |\alpha_l|^2 \boldsymbol{v}(f)\boldsymbol{v}^{\boldsymbol{H}}(f) + \sigma_w^2 \boldsymbol{I} = \mathbf{V}\mathbf{S}\mathbf{V}^H + \sigma_w^2 \boldsymbol{I} \tag{5}$$

where:

$$\mathbf{V}(f) = [1, \boldsymbol{v}(f_1) \ldots, v(f_{L-1})]^T \tag{6}$$

is an N×L matrix whose columns are the time-window frequency vectors from (3) at frequencies f_i *with* $i=0,\ldots,L$ of the complex exponentials and

$$\mathbf{S} = \begin{bmatrix} |\alpha_1|^2 & 0 & \cdots & 0 \\ 0 & |\alpha_2|^2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & |\alpha_L|^2 \end{bmatrix} \tag{7}$$

is a diagonal matrix of powers for each respective exponentials and autocorrelation matrix of the white noise will be:

$$\boldsymbol{R}_w = \sigma_w^2 \boldsymbol{I} \tag{8}$$

which is full rank, as opposed to $\mathbf{R}_s$ which was rank-deficient for $L<N$. In general, we will always choose the time window length, N to be greater than the number of complex exponentials L. The autocorrelation matrix in terms Eigen decomposition is

$$\boldsymbol{R}_x = \sum_{m=1}^{M} \lambda_m \boldsymbol{q}_m \boldsymbol{q}_m^{\boldsymbol{H}} = \mathbf{Q}\mathbf{D}\mathbf{Q}^H \tag{9}$$

where λ_m are the eigenvalues in descending order, that is, $\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_M$ and $\mathbf{q}_m$ are their corresponding eigenvectors. Here $\mathbf{D}$ is a diagonal matrix made up of the eigenvalues, and columns of $\mathbf{Q}$ are the related eigenvectors. The signal related eigenvalues can be written as the sum of the signal power and noise as follows:

$$\lambda_l = \boldsymbol{L}|\alpha_l|^2 + \sigma_w^2 \qquad for \quad l \le L \tag{10}$$

and the remaining eigenvalues are due to the noise only components, are

$$\lambda_l = \sigma_w^2 \qquad for \quad l > L \tag{11}$$

therefore, the Ļlargest eigenvalues correspond to signal made up of exponentials and the remaining eigenvalues have equal value and correspond to the noise. Thus, we can partition the correlation matrix into portions due to the signal and noise eigenvectors

$$\begin{aligned} \boldsymbol{R}_x &= \sum_{l=1}^{L}(M|\alpha_p|^2 + \sigma_w^2)\boldsymbol{q}_m\boldsymbol{q}_m^{\boldsymbol{H}} + \sum_{l=L+1}^{N}\sigma_w^2\boldsymbol{q}_m\boldsymbol{q}_m^{\boldsymbol{H}} \\ &= \mathbf{V}\mathbf{S}\mathbf{V}^H + \sigma_w^2\boldsymbol{I} = \mathbf{Q_s}\mathbf{D}\mathbf{Q_s^H} + \sigma_w^2\mathbf{Q_w}\mathbf{Q_w^H} \end{aligned} \tag{12}$$

where:

$$\mathbf{Q_s} = [\boldsymbol{q}_1, \boldsymbol{q}_2 \cdots, \boldsymbol{q}_L]\ ,\ \mathbf{Q_w} = [\boldsymbol{q}_{L+1}, \boldsymbol{q}_2 \cdots, \boldsymbol{q}_N] \tag{13}$$

are matrices whose columns consist of the signal and noise eigenvectors, respectively. The matrix **D** is L×L diagonal matrix containing the signal eigenvalues from (10). Thus, the N-dimensional subspace that contains the observations of the time-window signal vector from (6) can be split into subspaces spanned by the signal and noise eigenvectors, respectively. These two subspaces, known as the *signal subspace* and the *noise subspace*, are orthogonal to each other. Recall that the projection matrix from an N-dimensional space onto an *L*-dimensional subspace ($L<N$) spanned by vectors $\mathbf{Z}=[\mathbf{z}_1\ \mathbf{z}_2 \cdots \mathbf{z}_L]$ is

$$\mathbf{P} = \mathbf{Z}(\mathbf{Z^H}\mathbf{Z})^{-1}\mathbf{Z} \tag{14}$$

hence, the matrices that project a vector onto the signal and noise subspaces are as:

$$\mathbf{P_s} = \mathbf{Q_s}\mathbf{Q_s^H}\ ,\ \mathbf{P_w} = \mathbf{Q_w}\mathbf{Q_w^H} \tag{15}$$

since the eigenvectors of the correlation matrix are orthonormal then we have:

$$\mathbf{Q_s}\mathbf{Q_s^H} = \mathbf{I}\ ,\ \mathbf{Q_w}\mathbf{Q_w^H} = \mathbf{I} \tag{16}$$

since the two subspaces are orthogonal, then all the time-window frequency vectors from (3) must lie completely in the signal subspace, that is,

$$\mathbf{P_s}\boldsymbol{v}(f_i) = \boldsymbol{v}(f_i)\ ,\ \mathbf{P_w}\boldsymbol{v}(f_i) = \mathbf{0} \qquad for \quad 1 \le i \le L \tag{17}$$

However, in practice, the correlation matrix is not known and must be estimated from the measured data samples. If we have a time-window signal vector from (6), then we can form the data matrix by stacking the rows with measurements of the time-window data vector at a time *n* as follows:

$$\mathbf{X} = \begin{bmatrix} \boldsymbol{x}^T(0) \\ \boldsymbol{x}^T(1) \\ \vdots \\ \boldsymbol{x}^T(\mathrm{n}) \\ \vdots \\ \boldsymbol{x}^T(\mathrm{N}-2) \\ \boldsymbol{x}^T(\mathrm{N}-1) \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}(0) & \boldsymbol{x}(1) & \cdots & \boldsymbol{x}(\mathrm{L}-1) \\ \boldsymbol{x}(1) & \boldsymbol{x}(2) & \cdots & \boldsymbol{x}(\mathrm{L}) \\ \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{x}(\mathrm{n}) & \boldsymbol{x}(\mathrm{n}+1) & \cdots & \boldsymbol{x}(\mathrm{n}+\mathrm{L}-1) \\ \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{x}(\mathrm{N}-2) & \boldsymbol{x}(\mathrm{N}-1) & \cdots & \boldsymbol{x}(\mathrm{N}+\mathrm{L}-3) \\ \boldsymbol{x}(\mathrm{N}-1) & \boldsymbol{x}(\mathrm{N}) & \cdots & \boldsymbol{x}(\mathrm{N}+\mathrm{L}-2) \end{bmatrix} \tag{18}$$

which has dimensions of $N \times L$, where N is the number of data records or frames and L is the time-window length. From this matrix, we form an estimate of the correlation matrix, referred to as the sample correlation matrix

$$\hat{\boldsymbol{R}}_x = \frac{1}{N}\mathbf{X}^H\mathbf{X} \tag{19}$$

and the spectrum of ordered eigenvalues in (19), the "signal eigenvalues" are still identified as the largest ones. Conducting several computer simulations in the following section, we demonstrate that the proposed sparse sinusoidal model (FD-MSM) is a useful analysis model to reduce the feature dimensions while preserving the audio signal quality as close as possible to the original input.

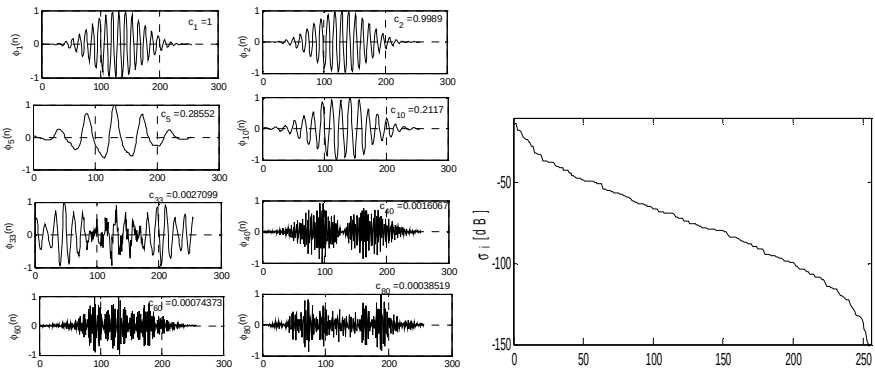
4 Simulation Results

Signal representation can be considered as a mapping from the N-dimensional space to a lower-dimensional feature space say L-dimensional space. Assuming a 1024-point FFT, and considering the symmetric property of FFT and removing the repeated part, we demonstrate that the target space i.e. sinusoidal space will have only a dimension varying between 20<L<40. The only constraint imposed to such sparse representation, is that it should preserve the input signal quality as close as possible.

4.1 PCA and Redundancy of Audio Signals

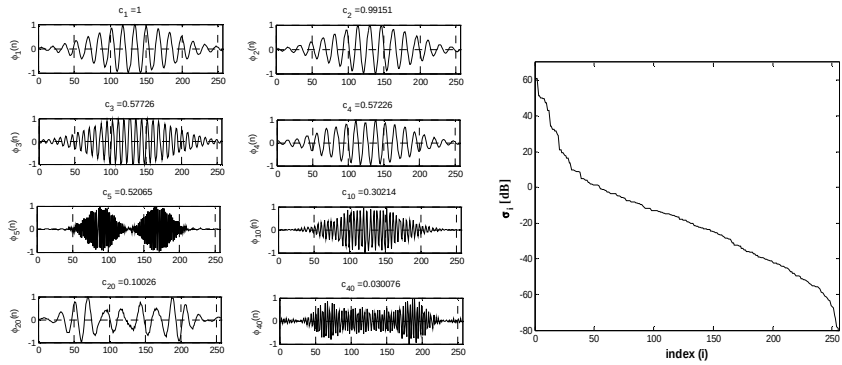
To show the redundancy of the FFT representation for audio signals, *Principle Component Analysis* (PCA) is performed on both speech and music. The procedure is treated as follows. The window length is set to 32 msec and 100 speech frames were ensemble averaged with a frame shift of 1 msec for stationarity assumption. Fig.1.a,b depict the eigen-values and eigen-vectors for several eigen-values, respectively. The number indicated in right-up section of each plot is related to corresponding eigen-value denoted by c_i where *i* is the eigen index.

As it is seen from Fig.1.a, the Eigen vectors obtained from *Singular Value Decomposition* (SVD) of frame covariance ensamples averaged matrix **R** determined in (19), the variance related to first few vectors are much notable. In contrast, ignoring vectors over i=33 results in an insignificant error (about 0.1%). In addition, Fig.4.b demonstrates the Eigen spectrum obtained from SVD decomposition in time-domain and STFT domain, respectively. As it is seen, the Eigen spectrum decreases monotonically in both cases very rapidly in that after about 30 index, the Eigen power attenuates to about -50 dB (=0.001%). In a similar manner for detecting correct model order



(a) (b)

Fig. 1. (a) Eigenvectrs as time domain basis and (b) eigen-values for a male speaker speech



(a) (b)

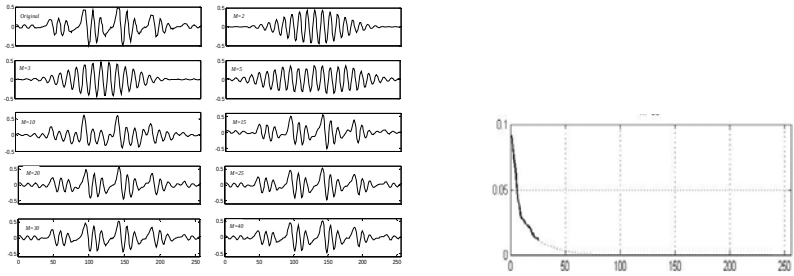
Fig. 2. (a) Eigenvectors as time domain basis functions and (b) eigen-values for a music signal

addressed, here by means of looking directly for a gap between the noise and the signal eigenvalues we observe the appropriate number of sinusoids. As a result, the FFT redundancy is obvious which in turn results in an imperfect signal representation especially for audio signals. Hence, translating speech signal to another domain with less dimension with respect to common STFT, we can expect a more compact and efficient representation which could be accomplished by sinusoidal signal representation discussed in this paper.

Simulation result for music are shown in Fig.2.a,b. Eigen-vectors are more sinusoid like than speech case. This inter correlation among eigen-vectors in time domain motivates us to employ a more compact representation. In addition, the steeply decrease in eigen-spectrum in Fig.2.b, verifies that 15-20 eigenvalues are considerable and the rest could be ignored without perceivable performance degradation.

4.2 Determining Number of Sinusoids Based on Eigen Decomposition

An experiment is conducted to confirm that the FD-MSM approach is in fact a sparse representation for audio signals. To proceed, a voiced frame is selected. Then sinusoidal analysis is performed for different number of sinusoids. Next, PCA is employed to obtain the fundamental eigenvalues. As shown in Fig.3.a and b, the number of prominent eigen-values related to signal subspace is $31<M'<35$ in harmony with frame reconstruction in Fig.3.a.



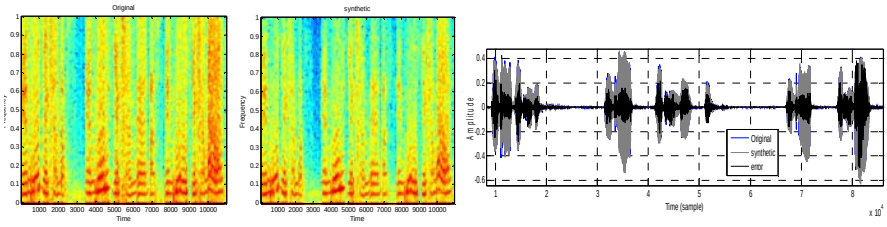
(a) (b)

Fig. 3. (a) Reconstruction a speech frame using different number of sinusoids (b) Eigen spectrum, choosing largest singular values for reconstruction (the dashed line) and the rest (in dotted line)

4.3 Subjective and Objective Results

Spectrograms and time signals are illustrated in Fig.4 both for the original input and the synthesized output signal. For subjective results, Mean Opinion Score (MOS) test [5] is conducted to measure the perceived quality for 8^{kHz} speech for 20 male and female speakers and their mixtures. 20 listeners were asked to score between 0-5 to reconstructed utterances. The MOS results are presented for different groups of listeners vs. the number of sinusoids, $M' \in [21,40]$ in Table.1. Waves can be downloaded at: http://ele.aut.ac.ir/pejmanmowlaee. It is observed that the proposed FD-MSM requires $25<M'<35$ to have negligible difference in MOS. Using $M'=33$ parameters are enough to establish trade-off between low dimensionality and high perceptual quality.

Comparing the subjective and objective results, we conclude that eigen-analysis results as shown in Fig.3.a,b propose using $31<M'<35$ sinusoids while MOS results indicate that a perfect reconstruction of speech signal is possible when $M' \approx 33$ which is in harmony with results in Fig.3.a. As a consequence, we opt for $M'=33$ to have an indistinguishable speech signal representation using the sinusoidal modelling. In addition evaluating FD-MSM for music signals, no distinguishable difference was inferred by listeners even for $M' \approx 14$. Evaluating these results with the conventional sinusoidal model proposed in [3],[4] they all need 5 to 10 more sinusoidal parameters than FD-MSM. In addition they all suffer from gross error related to pitch estimation while for analyzing audio mixtures as reported in [6]. In this case these methods are unable to extract the pitch value of the weaker speaker if the pitch value is equal to or a multiple of the pitch value of the stronger speaker and hence performance degrades.



(a) (b) (c)

Fig. 4. Showing spectrograms for (a) original, (b) synthesized, (c) related time domain signals

Table 1. MOS results for synthesized speech signals

M′	Category	MOS	M′	Category	MOS	M′	Category	MOS	M′	Category	MOS
25	*Adult male*	*3.8*	21	*Adult male*	*3.6*	33	*Adult male*	*4.4*	29	*Adult male*	*4.1*
	Child	*4*		*Child*	*3.85*		*Child*	*4.5*		*Child*	*4.3*
	Old Man	*3.9*		*Old Man*	*3.7*		*Old Man*	*4.4*		*Old Man*	*4.1*
	Old woman	*4.1*		*Old woman*	*4*		*Old woman*	*4.6*		*Old woman*	*4.5*
	Adult female	*4.1*		*Adult female*	*4*		*Adult female*	*4.6*		*Adult female*	*4.5*
40	*Adult male*	*4.6*	37	*Adult male*	*4.5*						
	Child	*4.75*		*Child*	*4.62*						
	Old Man	*4.65*		*Old Man*	*4.5*						
	Old woman	*4.8*		*Old woman*	*4.7*						
	Adult female	*4.8*		*Adult female*	*4.7*						

5 Conclusion

In this paper, proposed FDMSM model was studied for sparse representation of audio signals. We demonstrated that the space dimension was significantly reduced and the model can be assumed as a mapping from STFT spectrum space to a more compact one say 25<L<40 called sinusoidal subspace. This reduction is notable since no performance degradation in terms of perception and signal characteristic was. The choice of the number of sinusoids was confirmed using the state-of-the-art PCA.

References

1. O'Shaughnessy, D.: Speech Communications Human and Machine. IEEE press, NY (2000)
2. Macon, M.W., Clements, M.A.: Sinusoidal modeling and modification of unvoiced speech. IEEE Trans. Speech Audio Process 5(6), 557–560 (1997)
3. McAulay, R.J., Quatieri, T.F.: Speech analysis/synthesis based on a sinusoidal representation. IEEE Trans. ASSP 34, 744–754 (1986)
4. George, E.B., Smith, M.J.T.: Speech analysis/synthesis and modification using an analysis-by-synthesis/overlap-add sinusoidal model. Speech and Audio Processing, IEEE Trans. 5(5), 389–406 (1997)
5. Furui, S., Sondhi, M.M.: Advances in Speech Signal Processing. Marcel Dekker Inc., New York (1992)
6. Radfar, M.H., Dansereau, R.M., Sayadiyan, A.: Monaural speech segregation based on fusion of source-driven with model-driven techniques. Speech Comm. 49, 464–476 (2007)
7. Mowlaee, P., Sayadian, A.: A Fixed Dimension Modified Sinusoid Model (FD-MSM) for Single Microphone Sound Separation. In: International Conference on Signal Processing and Communications, ICSPC 2007, Dubai, United Arab Emirates, pp. 1183–1186 (November 2007)

Variant Combination of Multiple Classifiers Methods for Classifying the EEG Signals in Brain-Computer Interface

Zahra Shoaie Shirehjini, Saeed Bagheri Shouraki, and Maryam Esmailee

Department of Computer Engineering,
Sharif University of Technology, Tehran, Iran
Z_shoaie@ce.sharif.edu

Abstract. Controlling the environment with EEG signals is known as brain computer interface is the new subject researchers are interested in. The aim in such systems is to control the machine without using muscle, and we should control the machine using signals recorded from the surface of the cortex. In this project our focus is on pattern recognition phase in which we use multiple classifier fusion to improve the classification accuracy. We have applied various feature extraction methods and combined their results. Two methods, greedy algorithms and genetic algorithms, are used for selecting the pair feature extractor-classifier (we called expert) between the existed pair. Experiments show that with using some combination method such as majority vote, product, mean, median we have obtained better result than best existing result and Fuzzy integral method and decision template have shown the similar result with the best result in BCI competition 2003 [15].

Keywords: EEG signal, classification, combination of multiple classifiers, feature extraction, majority voting, fuzzy measure and integral, decision template.

1 Introduction

A brain computer interface (BCI) provides an alternative communication channel between a user's brain and a device. A successful BCI design enables its users to control their environment (e.g., light switch or a wheelchair), a neural prosthesis or a computer by thinking of it only. This is done by measuring specific features of a person's brain signal that relate to his/her intent to affect control. These features are then translated into signals that are used to control/actuate devices. For a review of the field, see [1]. In this paper we use variant famous methods of feature extraction for extracting features that we can analyze signals in different aspect. For example for more nauseating signals, fractal dimension is a appropriate feature and for signal with less turbulent, PSD is a good feature for feeding to a classifier [2].Feature extraction methods that we used do not belong to special mental activity and they were methods that we expect they can have appropriate results in classifying EEG signals. Then we used SVM classifier that has good performance in classifying of EEG signals for BCIs. The reason for this, is the concept to using "simple method first" and the fact that in our BCI studies linear classification methods were rarely found to perform

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 477–484, 2008.

worse than non-linear classifier [9, 10, 11, 13]. For every feature extraction method we found the optimal parameter of SVM (∂, c), and we called the feature extractor-classifier pair as an expert that must use for combination state. In combination state we use two algorithms for selecting the pairs contribute in the combination. First with greedy algorithm we pick the expert with less error percentage for combination stage, second we use genetic algorithm for select the pairs participate in the combination. These methods were operated on Data set provided by Department of Medical Informatics, Institute for Biomedical Engineering, University of Technology Graz. (Gert Pfurtscheller). This dataset was recorded from a normal subject (female, 25y) during a feedback session. The subject sat in a relaxing chair with armrests. The task was to control a feedback bar by means of imagery left or right hand movements. The experiment consists of 7 runs with 40 trials each. Given are 280 trials of 9s length. The first 2s was quite, at t=2s an acoustic stimulus indicates the beginning of the trial, and a cross "+" was displayed for 1s; then at t=3s, an arrow (left or right) was displayed as cue. At the same time the subject was asked to move a bar into the direction of the cue [14]. Experimental results have shown that we can improve the classification accuracy using combination of classifier if we use appropriate expert beside each other. In this research we reach to 92.14% accuracy that develops the best result in BCI competition 2003 about 2.85% [15].

The organization of the paper is as follows. Section 2 introduces the feature extraction methods. Section 3 is about the base classifier that we use in this research. In section 4 combination of multiple classifier methods will be initiated. Finally, we conclude our paper and discuss some future work.

2 Feature Extraction

The Features need to reflect properties of EEG that are relevant for the recognition of mental activities. The choice of adequate features to characterize EEG has been the object of active research during the last decades [16, 17].

We used variant feature extraction method for analyzing signal from different characteristic. The methods that we used were applied for extracting features in many previous researches and have appropriate result in classifying. These methods in concern to theirs nature is categorized as a below:

1-parametric feature,2- Frequency feature,3-Statistical feature,4-Time-frequency feature,5-Entropy feature, 6-fractel features

3 Basic Classifier

Basic classifier that we used in this paper is SVM. Support vector machine (SVM) has been widely used in pattern recognition and regression due to its computational efficiency and good generalization performance. It was originated from the idea of the structural risk minimization that was developed by Vapnikin 1970's [20].

Let $E = \{e_1, e_2, , , e_k\}$ be a set of trained classifiers (called also ensemble, team, pool, etc.), and $C_\Lambda = \{C_1, C_2, ..., C_M\}$ be a set of class labels. Each classifier gets as its input a feature vector $\mathcal{X} \in \Re^n$ and assigns it to a class label from C_Λ, i.e.,

$D_i : \Re^n \to C_\Lambda$ or equivalently, $D_i(x) \in C_\Lambda$, $i = 1,2, \ldots M$ alternatively, the classifier output can be formed as a M-dimensional vector

$$D_i(x) = [d_{i,1}(x), \ldots d_{i,M}(x)]^T \quad (1)$$

Where $d_{i,j}(x)$ is the degree of "support" given by classifier D_i to thce hypothesis that x comes from class C_j . Most often $d_{i,j}(x)$ is an estimate of the posterior probability $P(c_i|x)$.

It is convenient to organize the output of all $\mathcal{L}$ classifiers in a decision profile.

$$DP(x) = \begin{bmatrix} d_{1,1}(x) & \ldots & d_{1,j}(x) & \ldots & d_{1,M}(x) \\ d_{i,1}(x) & \ldots & d_{i,j}(x) & \ldots & d_{i,M}(x) \\ d_{L,1}(x) & \ldots & d_{L,j}(x) & \ldots & d_{L,M}(x) \end{bmatrix} \quad (2)$$

Thus, the output of classifier D_i is the i-th row of the decision profile, and the support for class C_j is the j-th column. In [24] methods for calculating the probability from the output of SVM classifier was conversed.

4 Combination of Multiple Classifiers

Difficult pattern recognition problems involving high dimensional patterns, large numbers of classes, and noisy inputs can be solved efficiently using systems of multiple classifiers. The combination of multi classifiers can be considered as a generic pattern recognition problem in which the input consists of the result of individual classifiers, and the output is the combined decision [3].

It is based on the idea that classifiers using different methodologies or different features can complement each other in classification performance and increase the probability that the errors of the individual features or classifiers maybe compensated by the correct results of the rest. This has led to a belief that by using features and classifiers of different types simultaneously, classification accuracy can be improved [5] such that the performance of the combination is never worse than the average of the individual classifiers, but not necessarily better than the best classifier [6].

The methods that can be used to combine multiple classifier decision depend on the type of information produced by the individual classifiers. As we said the classifiers produced information in the form of either: a single class label indicating that this class has the highest probability to which the input pattern belongs; or certainty measure values being assigned to each class label indicating the degree that the corresponding class pertains to the pattern.

The combining methods that use the whole posterior probability matrix as input is called class- independent method, and the methods that use the columns of the posterior probability matrix as input is called class-dependent method [21].These two approach are applied on data and result of them are exist in section 5. In section A and B we briefly discussed these methods.

4.1 Class-Dependent Method

Class-independent combiners use the column of posterior probability matrix ($DP(x)$) with use of

$$\mu_j(x) = Rule\left(d_{1,j}(x), \cdots d_{i,j}(x), \cdots d_{L,j}(x)\right) \tag{3}$$

We look for a vector with M final degrees of support for the classes as a soft label for x, denoted

$$D(x) = [\mu_i(x), \cdots, \mu_M(x)]^T \tag{4}$$

If a single (crisp) class label of x is needed, we use the maximum membership rule: Assign x to class C_j iff

$$\mu_s(x) \geq \mu_t(x), \forall t = 1, \cdots, M \tag{5}$$

For making decision, two types of combination methods (rule) have been investigated: one used fixed combination rules that don't require prior training such as mean, max, min, median, product, majority vote etc, and a trainable combination method in which the outputs of the base classifiers where used as the input features of a general classifier used for classifier fusion. For the second type, the fuzzy integral method was used wherein the combination operator also function as a classifier, where a training set is used to adapt the combining classifier to the classification problem. In next section we briefly initiate fuzzy integral combiner.

Fuzzy integral

Let $C_\Lambda = \{C_1, C_2, \cdots, C_M\}$ be the set of classes into which patterns will be classified. Let $E = \{e_1, e_2, \cdots, e_k\}$ be the set of classifiers, and x be the pattern under consideration for classification. Let $H(e_k): E \rightarrow [0,1]$ be the certainty set of classifier e_k containing the partial evolution of the pattern x for classes set C_Λ, i.e., $H(e_k) = \{h_1 e_k, h_2 e_k, \cdots, h_M e_k\}$, such that $h_1 e_k$ is an indication of the certainty of pattern x classification to class C_i using classifier e_k, where 1 indicates absolute certainty that pattern x belongs to class C_i and 0 implies absolute certainty that it does not belong to class C_i.

Before defining how FI combines information sources, let's look to a conventional weighted arithmetical mean (WAM) operator [22]. A final support measure for class Cj using WAM c

$$M_{WAM} = \sum_{i \in L} \mu(i) h_i$$
$$\text{Where } \textstyle\sum_{i \in L} \mu(i) = 1 \text{ (additive)}, \mu(i) \geq 0 for\ all\ i \in L. \tag{6}$$

and be defined as The WAM operator combines the score of $\mathcal{L}$ competent information sources through the weights of importance expressed by μ_i . The main disadvantage of the WAM operator is that it implies preferential independent of the information source [22].

Let's denote with $\mu(\mathrm{i},\mathrm{j}) = \mu\{(\mathrm{i},\mathrm{j})\}$ the weight of importance corresponding to the couple of information source i and j from $\mathcal{L}$. If μ is not additive, i.e. $\mu(\mathrm{i},\mathrm{j}) \neq \mu(\mathrm{i}) + \mu(\mathrm{j})$ for a given couple $(\mathrm{i},\mathrm{j}) \subseteq \mathcal{L}$, we must take into account some interaction among the information source. Therefore, we can build an aggregation operator starting from the WAM, adding the term of "second order" that involve the corrective

coefficients $\mu(i,j) - \mu(i) + \mu(j)$,then the term of "third order",etc. In this way, we arrive to the definition of the FI: assuming the sequence $h_i, i = 1, \ldots \mathcal{L}$, is ordered in such way that $h(e_1) < \cdots h(e_k)$, the Choquet fuzzy integral [22, 3, 4] can be computed as

$$M_{FI}(\mu) = \sum_{k=1}^{L}[\mu(k,\cdots,L) - \mu(k+1,\cdots,Z)]h_j(e_k) \quad (7)$$

Where $\mu(k,\cdots,L) = \mu(\emptyset) = 0.$

is called fuzzy measure (FM). Thus, $\mu(s)$ is a weight related to subset S of the set $\mathcal{L}$ of information sources.

Corresponding to each classifier e_k , the degree of confidence, $\mu^{i/k}$, of how accurate classifier e_k is in the recognition of the class C_i must be given. The degree of confidences, $\mu^{i/k}$ is called fuzzy densities and can be subjectively assigned by an expert, or determined via some statistical measurements on a training set.

As starting step in this research, a simple method to estimate the densities $\mu^{i/k}$ was used. These values were selected to be proportional to the correct classification rates of each classifier. Consider the confusion matrix of classifier e^k denoted as a $C(e_k)$, which contains the results of correctly classified and misclassified patterns. It was constructed for each classifier and expressed in the form:

$$C(e_k) = [c_{ij}^k] \quad (8)$$

Where i=1,2,..,M, j=1,2,..,M+1, and M is the number of classes. For i=j, c_{ij}^k is the number of correctly classified patterns in class C_j by classifier e_k. C^i being misclassified as class C^j by classifier e^k . The fuzzy density values were defined as:

$$\mu^{i/k} = c_{ii}^k / \sum_{j=1}^{M} c_{ij}^k \quad (9)$$

Once the $\mu^{i/k}$ s were evaluated, the λ-fuzzy measures, $\mu_\lambda(A_k)$, where $A^k = \{e^1, e^2 ..., e^k\}$, were constructed for each class recursively from:

$$\begin{aligned} &\mu_\lambda(A_1) = \mu_\lambda([e_1]) = \mu^{i/1}, \quad for 1 \le i \le M \\ &\mu_\lambda(A_k) = \mu^{i/k} + \mu_\lambda(A_{k-1}) + \lambda_i \mu^{i/k} \mu_\lambda(A_{k-1}) \\ &\qquad for 1 < k \le K \quad and \quad 1 \le i \le M \end{aligned} \quad (10)$$

λ_i was obtained using formula ():

$$\lambda_i + 1 = \prod_{k=1}^{K}(1 + \lambda_i \mu^{i/k}), \quad for 1 \le i \le M \quad (11)$$

This was calculated by solving the (K-1) degree polynomial and finding the unique root greater than -1.

The overall confidence for the class was the fuzzy integral value calculated using the Sugeno fuzzy integral with respect to fuzzy measure μ_λ over E [7]:

$$S_g(h) = \int_A h(e).\mu(.) = \max_{k=1}^{K}[\min(h(e_k, \mu(A_k)))] \tag{12}$$

Or using the Choquet fuzzy integral [8]:

$$C\mu(h) = \sum_{k=1}^{K} h(e_k)[\mu(A_k) - \mu(A_{k+1})] \tag{13}$$

$$taking \quad \mu(A_{k+1}) = 0$$

Actually, FI can be seen as a compromise between the evidence expressed by the outputs of the classification systems and the competence represented by the FM [21], and in addition to FM that provides an initial view about the importance of information source , all possible subsets of Z that include that information source should be analyzed to give a final score.

A fuzzy integral distinguishing characteristic is that utilize information concerning the worth or confidence in subsets of information sources in the decision making process [18] represented by a fuzzy measure. In the classifier fusion process, fuzzy integral combine objective evidence, supplied by the classifiers in the form of certainty measures, for a hypothesis with the prior expectation of the worth (fuzzy density values) of subsets of these classifiers.

4.2 Class-Independent Method

The combining methods that use the whole posterior probability matrix as input is called class independent method. In this category we used decision template method [21] as combiner that it is trainable, simple and intuitive rule.

The idea of the decision template (DTs) model is to “remember “ the most typical decision profile for each class, called the decision template for that class, and then compare it with the current decision profile . The closest match will label x. Different similarity measure can be used [23].In this paper we used Euclidean distance and Mahalanobis as similarity measure.

5 Simulation and Results

For selecting experts (feature extractor-classifier) within existed experts, we use two methods, greedy algorithm and genetic algorithm. In greedy method the experts with minimum error chose for combination stage. In second method we used genetic algorithm in which we chose the binary chromosome that 1 indicates being experts and 0 shows not existence of experts in combination step. We use the final classification accuracy as fitness function.

When we use single classifier best result obtained with PSD feature extraction method that has %11.25 error. Table 1 show the error obtained with classifier fusion when we use two described method, in which DT1 refer to decision template with Euclidean distance for similarity measure and DT2 refer to decision template with Mahalanobis distance for similarity measure. We can conclude from the obtained results that in spite of what expected before, combination methods always don’t

improve the accuracy of classification, and in some case the result may be worth than the single classifier. In other world the final classification accuracy directly depends on the experts that we chose for combination. As the table 1 show when we used the suitable experts beside each other (expert selected with genetic algorithm) we could gain the minimum error that progressed the best result in BCI competition 2003 about 2.85%.

Table 1. Simulation Result With Combination Method And Two (Genetic And Greedy) Algorithm

Combination rule	Error% (Genetic)	Error% (Greedy)
Max	%10.71	%15.71
Min	%11.43	%15.71
Mean	%9.29	%15
Median	%9.29	%14.29
product	%8.57	%15
Majority Vote	%7.86	%12.14
Sugeno-Fuzzy Integral	%11.43	%14.29
Choquet-Fuzzy Integral	%10.71	%15.71
Decision Template1	%10.71	%26.43
Decision Template2	%10.71	%15.71

6 Discussion

In this paper we have been using some of feature extraction methods and SVM classifier. The next step would be to provide some nonlinear classifiers and complicated feature extraction and feature classification methods in order to build a more powerful system and try this model for a multi-class problem and by the way with some algorithms, for instance Validation methods we can choose some appropriate features that satisfies complimentary information property and have good performance for classifying.

References

1. Wolpaw, J.R., Birbaumer, N., McFarland, D.J., Pfurtscheller, G., Vaughan, T.M.: Brain-computer interfaces for communication and control. Clin. Neuro. 113(6), 767–791 (2002)
2. Boostani, R.: EEG Classification IN Brain Computer Interface, Doctoral thesis, Amirkabir University of Technology (2005)
3. Sugeno, M.: Theory of fuzzy integrals and its applications, Doctoral thesis, Tokyo Institue of Technology (1974)
4. Keller, J.M., Gader, P., Tahani, H., Chiang, J.H., Mohamed, M.: Advances in fuzzy intelgration for pattern recognition. Fuzzy Sets and Systems 65, 273–283 (1994)
5. Kamel, M.S., Wanas, N.M.: Data dependence in combining classifiers. In: Windeatt, T., Roli, F. (eds.) MCS 2003. LNCS, vol. 2709, pp. 1–14. Springer, Heidelberg (2003)
6. Stashuk, D.W., Paoli, G.M.: Robust supervised classification of motor unit action potentials. Medical & Biological Engineering & Computing 36(1), 75–82 (1998)
7. Tahani, H., Keller, J.M.: Information fusion in computer vision using the fuzzy integral. IEEE Transactions on systems, Man, and Cybernetics, Vol 20(3), 733–741 (1990)
8. Duin, R.P.W.: The combining classifier: to train or not to train? In: Processing of 16th International Conference on Pattern Recognition, vol. 2, pp. 765–770 (2002)

9. Parra, L., Alvion, C., Tang, A.C., Pearlmutter, B.A., Yeung, N., Osman, A., Sajda, P.: Linear spatial integration for single trial detection in encephalography. NeuroImage 7(1), 223–230 (2002)
10. Muller, K.R., Anderson, C.W., Birch, G.E.: Linear and non-linear methods for brain-computer interfaces. IEEE Trans. Neural Sys. Rehab. Eng. 11(2), 165–169 (2003)
11. Keirn, Z.A., Aunon, J.I.: Man-Machine Communications Through Brain-Wave Processing. IEEE Engineering in Medicine and Biology Magazine 9(1), 55–57 (1990)
12. Garret, D., Peterson, D.A., Anderson, C.W., Thaut, M.H.: Comparison of Linear,Nonlinear, and Feature Selection Methods for EEG Signal Classification. IEEE Transactions on Neural Systems and Rehabilitation Engineering 11(2), 141–144 (2003)
13. Garcia, G.N., Ebrahimi, T., Vesin, J.-M.: Classification of EEG signals in the ambiguity domain for brain computer interface applications. In: Proceedings of the IEEE International Conference on Digital Signal Processing (DSP), vol. 1, pp. 301–305 (July 2002,)
14. Blankertz, B., Muller, K.R., Curio, G., Vaughan, T.M., Schalk, G., Wolpaw, J.R., Schlogl, A., Neuper, C., furtscheller, G.P.: The BCI competition 2003: Progress and perspectives in detection and discrimination of EEG single trials. IEEE Trans. Biomed. Eng. (to appear, 2004)
15. Blankertz, B.: BCI Competition 2003 results (web page)
16. Niedermeyer, E., da Silva, F.H.L.: Electroencephalography: Basic Principles, linical Applications and Related Fields, 4th edn. Williams and Wilkins (1999)
17. Windhorst, U., Johansson, H.: Modern Techniques in Neuroscience Research. Springer, Heidelberg (1999)
18. Keller, J.M., Osborn, J.: Training the fuzzy integral. International Journal of approximate Reasoning 15(1), 1–24 (1996)
19. Rasheed, S., Stashuk, D., Kamel, M.: Multi-Classification Techniques Applied to EMG Signal Decomposition. In: 2004 IEEE International conference on systems, Man and Cybernetics (2004)
20. `http://www.cs.cmu.edu/~awm/tutorials`
21. Kuncheva, L.: Fuzzy versus nonfuzzy in combining classifiers designed by boosting. IEEE Transaction on fuzzy system 11(6) (December 2003)
22. Marichal, J.l.: An axiomatic approch of the discrete Choquet integral as a tool to aggregate interacting criteria. IEEE Transactions on fuzzysystem 8 (2000)
23. Bouchon-Meunier, B., Rifiqi, M., Bothorel, S.: Toward general measures of comparision of objects. fuzzy sets system 84(2), 143–153
24. Platt, J.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods

Nevisa, a Persian Continuous Speech Recognition System

Hossein Sameti, Hadi Veisi, Mohammad Bahrani, Bagher Babaali, and Khosro Hosseinzadeh

Department of Computer Engineering,
Sharif University of Technology, Tehran, Iran
sameti@sharif.edu
{veisi,bahrani,babaali,hosseinzadeh}@ce.sharif.edu

Abstract. In this paper we have reviewed Nevisa Persian speech recognition engine. Nevisa is an HMM-based, large vocabulary speaker-independent continuous speech recognition system. Like most successful recognition systems, MFCC with some modification has been used as speech signal features. It also utilizes a VAD based on signal energy and zero-crossing rate. Maximum likelihood estimation criterion the core of which are the classical segmental k-means and Baum-Welsh algorithms is used for training the acoustic models. The system is based on phoneme modeling and utilizes synchronous beam search based on lexicon tree for decoding the acoustic utterances. Language modeling for Persian has been implemented in two statistical (n-gram) and grammatical forms. Nevisa is equipped with out-of-vocabulary capability for applications with small size vocabulary. In order to compensate the effect of accuracy reduction in noisy environments, powerful robustness methods are utilized. Model-based approaches like PMC, MLLR, and MAP, feature robustness methods like CMS, PCA, RCC, and VTLN, and speech enhancement methods like spectral subtraction and Wiener filtering were investigated. Some of these methods were modified to achieve higher robustness. For training Nevisa, Farsdat database was used. To evaluate the system accuracy, a clean test set was selected from Farsdat and four noisy tasks with different noise types were recorded in different real environments. By taking the advantages of the robustness methods, performance of Nevisa in real environments is similar to clean condition.

Keywords: Continuous Speech Recognition, Persian (Farsi) language, Nevisa, language modeling, search and decoding, robustness, grammar.

1 Introduction

Speech is the easiest and most natural way of communication for human beings. But natural speech varies in many aspects. Not only different speakers have different voices, but also there are substantial variations in the voice of single speaker. In fact there are no two identical speech signals even from a single speaker. Also practical acoustic conditions are usually noisy conditions where speech signals are distorted. Despite the complexities of automatic speech recognition and perception, recent progresses in signal processing, classification, search methods, and language modeling have attained admissible results in solving this problem. Having a speech recognition

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 485–492, 2008.

system which successfully maps a continuous speech signal to related discrete symbols is not a dream anymore. Many successful systems have emerged in the last two decades [1,2,3] which have achieved acceptable ability in performance. Nowadays, speech recognition is not just an academic and research subject but a technology resulting in practical systems for real applications. Many speech recognizers have achieved many necessary and beneficial features such as high accuracy, speaker independency and real-time recognition. They are equipped with linguistic information and noise robustness methods. Beside all of these advances the speech recognition systems are not indeed universal and maximally beneficial. They are limited and lack enough flexibility in speaking style and speaker's accent; they cannot use semantic and pragmatic information as humans and their sensitivity to noise is not resolved completely.

There are different approaches in automatic speech recognition (ASR) but successful speech recognizers utilize pattern recognition approach which consists of statistical and artificial intelligence (AI) methods [1,3,4]. These systems are made up of separable blocks such as acoustic front end, acoustic modeling, language modeling and hypothesis search. The front end block preprocesses the speech signals and transforms it into features vectors. There are various approaches to feature extraction but cepstrum-based methods such as Mel-Frequency Cepstral Coefficients (MFCC) are used commonly [4]. Typical methods for modeling the acoustic features are artificial neural networks (ANN) [1, 4] and hidden Markov models (HMM) [3, 4, 5]. Statistical (n-grams) and structural models are often used as linguistic information [4, 6]. Final goal of developing speech recognizers is putting them in practical use but their performance degrades dramatically in real acoustic conditions where the speech signal is deformed due to noise interference. Robustness techniques are used to compensate this effect and retain the performance level of these systems as in clean conditions.

The most progresses in ASR systems have been achieved on English and a few other languages [1, 2]. In recent years, speech recognition for Persian language has been addressed by some Persian researchers [7-10]. This paper describes Nevisa Persian speech recognition system. Nevisa is an efficient speaker independent large vocabulary continuous speech recognizer that utilizes many of the recent advancements in ASR. It is a modular and flexible system that can be used in many applications and research works easily. This system was first introduced in [7, 8] as Sharif speech recognition system. It employs the cepstral coefficients as acoustic features and continuous density hidden Markov models (CDHHM) as acoustic model [4, 5]. A time-synchronous left-to-right Viterbi beam search in combination with a tree-organized pronunciation lexicon is used for decoding [11, 12]. To constrain the search space, two pruning techniques are employed in the decoding process. Due to our practical approach in using this system, Nevisa is equipped with most popular speaker environmental noise robustness techniques. Various data compensation and model compensation methods are used to achieve this objective. Also class-based n-gram language models (LM) [13, 14] with GPSG-based Persian grammar [15] are utilized as word-level and sentence-level linguistic information. In the remainder of this paper, an overview of Nevisa Persian speech recognition system and overall features of this system is given in Section 2. Section 3 describes the framework of experiments and provides some recognition results of Nevisa in clean acoustic condition and in real applications in noisy environments. Finally, Section 4 gives a brief summary and conclusion.

2 Nevisa Speech Recognition Systems: Overview

Nevisa is a CDHMM-based speech recognition system using MFCC signal representation. The overall architecture of the system is shown in Fig. 1. Each labeled block in this Figure represents a module that can be easily changed or replaced. This makes the system very flexible in developing various applications or experimenting different module implementations in research works. Dashed blocks are robustness modules that can be used optionally. These modules and their methods will be discussed in the coming sections. Voice activity detector (VAD) is a useful block in ASR systems in real applications to detect speech parts from non-speech signal parts. Nevisa uses energy and zero-crossing based VAD in the pre-processing of the speech signal. This unit specifies the beginning and the end of utterance and reduces the processing cost of the feature-extraction and recognition/decoding blocks. The modified VAD is also used in spectral subtraction (SS) and the introduced PC-PMC robustness methods to detect noise segments in the speech signal. The MFCC is used as the core of the feature extraction block which is supplied with Vocal Tract Length Normalization (VTLN [17]), Cepstral Mean Subtraction (CMS) and Principal Component Analysis (PCA) robustness methods. In addition to speech enhancement and feature robustness

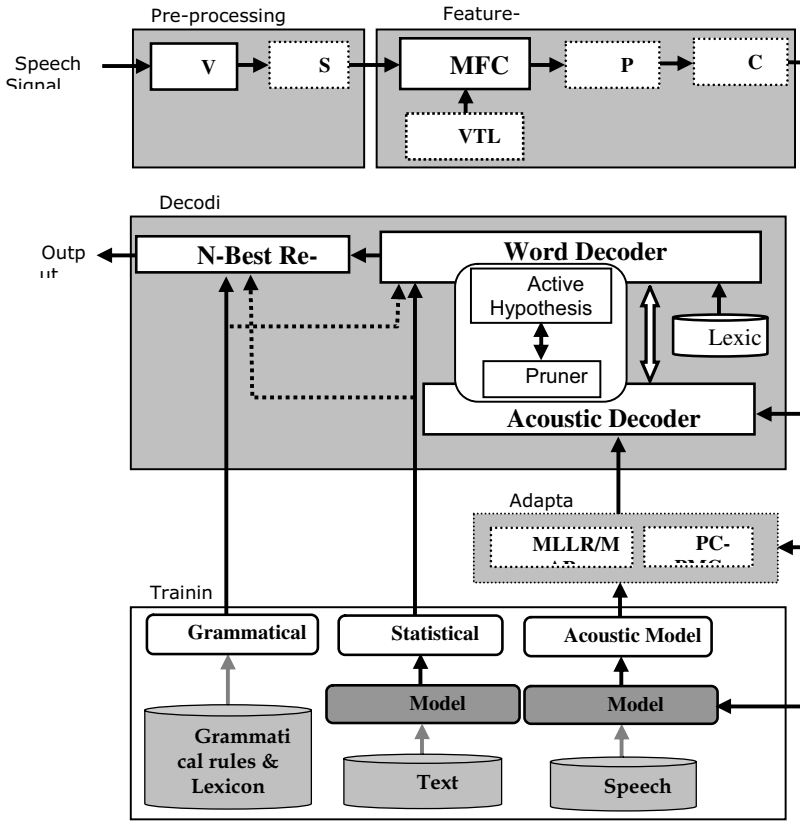


Fig. 1. The architecture of Nevisa

techniques, MLLR, MAP [19] and PC-PMC model adaptation methods can be applied optionally on acoustic models to modify the acoustic model parameters and adapt to speaker and environmental variations.

This system uses mono-phone acoustic models that are represented by continuous density hidden Markov models. These models are mixtures of Gaussian distribution in cepstral domain. Forward, skip and loop transitions between the states are allowed. Covariance matrices are modeled by a single diagonal matrix for each Gaussian distribution [7, 10]. The parameters of the emission probabilities are trained using the maximum likelihood criterion. The training procedure is initialized by a linear segmentation. Each iteration of the training procedure consists of time alignment by dynamic programming (Viterbi algorithm) followed by parameters estimation, resulting in segmental k-means training procedure [3, 4]. In decoding phase, a Viterbi-based search with beam and histogram pruning techniques is used. In this module the recognized acoustic units are used to make active hypotheses via word decoder. The word decoder searches the lexicon tree simultaneously in interaction with acoustic decoder and pruner modules. The final active hypotheses are rescored using language models before making the output. Both statistical and grammatical language models can be used either in word decoder or in rescore modules. In Nevisa, by default, statistical LM is used in the word decoder i.e. during the search, and the grammatical model is used in N-best rescore module. Dashed arrows in Fig 1 means that statistical LM can be used in rescorer module and grammatical LM can be utilized during the search.

3 Experiments

3.1 Tasks and Databases

In training the Nevisa, hand-segmented Farsdat database [20] has been used. This database contains 6080 Persian utterances, uttered by 304 speakers. Each speaker has uttered 18 randomly chosen sentences (from a set of 405 sentences) plus two sentences which are common for all speakers. The sentences are formed by using over 1000 Persian words. The speakers are chosen from 10 different geographical regions in Iran so it includes 10 most common dialects of the Persian language. Male to female population ratio is 2 to 1. The database is recorded in a low noise environment featuring 31dB signal to noise ratio in average. Our clean test set is selected form this database and contains 140 sentences from 7 speakers. All of the other sentences are used as train set. We have used "Persian Text Corpus" as discussed above for statistical language modeling.

To evaluate the performance of Nevisa in the real applications and in noisy environments, FArsi NOisy Speech (FANOS) database version 1.0 is recorded and transcribed [16, 18]. This database consists of 4 pair sets providing 4 tasks. As adaptation techniques are used in robustness methods, each task in this database includes two subsets identified as adaptation subset and test subset. Each adaptation subset is arranged as follows: 175 sentences (selected from Farsdat sentences) are uttered by 7 speakers including 5 males and 2 females. Each speaker reads 10 common sentences (read by all speakers) plus 15 different sentences. Also each test subset consists of

140 sentences uttered by 7 speakers including 5 males and 2 females, each reading 20 sentences. The average length of the sentences is 3.5 seconds. The transcriptions are at word level for test data and at phoneme level for adaptation data. Each task demonstrates a new environment which differs from the training environment. Tasks A and B are recorded in office environment with condenser and dynamic microphones respectively with average SNRs of 12dB and 30dB.Both tasks C and D are recorded with condenser microphone in office environment and in presence of exhibition and car noises respectively. Corresponding SNR levels of these sets are 9dB and 7dB. Table 1 summarizes the properties of the tasks in the FANOS database.

Table 1. The characteristics of tasks in FANOS 1.0 database

	Environment	**Microphone**	**SNR(dB)**	**No. of speech files (adaptation + test)**	**No. of speakers (male + female)**
Task A	Office	Condenser	12	315 (175 + 140)	7 (5+2)
Task B	Office	Dynamic	30	315 (175 + 140)	7 (5+2)
Task C	Exhibition	Condenser	9	315 (175 + 140)	7 (5+2)
Task D	Car noise	Condenser	7	315 (175 + 140)	7 (5+2)

3.2 System Parameters

In the acoustic front-end the speech is sampled at 22050 kHz and is blocked into 20-ms frames of overlapped by 12ms. A Hamming window is also applied to the signal in order to reduce the effect of frame edge discontinuities. After that a 1024-point Fast Fourier Transform (FFT) is performed. The magnitude spectrum is warped according to the signals warping factor if the VTLN option used. The obtained spectral magnitude spectrum is integrated within 25 triangular filters arranged on the Mel-frequency scale. The filter output is the logarithm of sum of the weighted spectral magnitudes. Then discrete cosine transform (DCT) is applied that results 12 cepstral coefficients. First and second derivatives of cepstral coefficients are calculated using linear regression methods [16] over a window covering 7 neighboring cepstrum vectors. This makes up vectors of 36 coefficients per speech frame. Finally, PCA and/or CMS are used if these options are to be used.

Nevisa uses HMM modeling, each HMM representing one of the 29 phonemes Of the Persian language. Also, we have used additional model for representing silence. All HMMs are left-to-right and they are composed of 6 states and a mixture of 16 Gaussians per state. Forward and skip transitions between the states and self-loop transitions are allowed. The elements of the feature vectors are assumed uncorrelated resulting in diagonal covariance matrices. The initialization of parameters is done using linear segmentation and then the segmental k-means estimates the expected parameters after 10 iterations. The beam width in the decoding process is 70.0 and the stack size is 300. The vocabulary contains 1092 words that includes all words appeared in FARSDAT database.

3.3 Results

Table 2 summarizes the results of Nevisa system on clean test set using word error rate (WER) as the evaluation criteria. As mentioned in section 6.1, the test set contains 140 sentences from 7 speakers. In the results reported here, the Witten-Bell smoothing technique [13] is used. As the results show, the base-line (BL) with no language model results relatively high WER but the linguistic information improve the system performance. In these results, class-based bi-gram with 200 classes reduces the perplexity noticeably but the perplexity of the test set is relatively high and this is because of the nature of sentences in Farsdat database. In this database, sentences are made artificially to cover all acoustical variations in Persian language. The reduction in WER obtained by using the grammar is noticeable but since the grammar is used in the end of the search, the perplexity has not been reduced.

Table 2. Performance of Nevisa on word level in clean acoustic condition

	WER%	**Perplexity**
BL-No LM	38.14	-
POS-Based Bi-gram	24.68	2105
Class-Based Bi-gram	23.4	1045
POS-Based Bi-gram Grammar	18.2	2105

The performance of Nevisa is degraded in adverse noisy condition as for all other recognition systems. Equipping this system with various compensation methods has made it robust to different noise types. Table 3 shows the recognition results of the system on four noisy tasks. The POS-based bi-gram language model has been used in the evaluations in this table. The perplexities of all of noisy tasks are the same since they use identical sentences. The sentences of these tasks are similar to the sentences of clean test set and so, the perplexity of noisy tasks are almost the same as the perplexity of clean task mentioned in table 3.

As these results show, the WER of the system on these tasks are very high. The recognition rates on task C and task D are negative due to the high insertion rate in these conditions. Using speaker and environment compensation methods the performance of the system is considerably improved. Each of the robustness methods mentioned in section 5 provides better recognition rate in comparison with the base-line system and some combinations of these methods result in even higher performances. Some examples of the compensation ability of the robustness methods are shown in the table 4. Clearly, they have provided obvious enhancement in the recognition accuracy of the system. We can see that VTLN provides good compensation for less-noisy environments like task A and task B while PMC and PC-PMC result in better compensation in more noisy environments. In the PC-PMC method, numbers of features are reduced 25% from 36 to 25. MLLR and MAP adapt the acoustic models to environmental conditions, microphone and speaker's signal properties. MAP has good ability in adaptation when the adaptation data is enough and MLLR provides better

Table 3. Evaluation of Nevisa and the robustness methods on FANOS noisy tasks (WER% on word level)

Robustness Method	Task A	Task B	Task C	Task D
None	74.04	75.32	116.41	105.94
VTLN+MLLR	30.37	32.87	82.52	60.07
PMC-MAP	38.63	50.49	69.36	50.22
PC-PMC+MLLR	31.33	28.70	56.17	42.11

adaptation in less-noisy conditions compared with noise-dominant conditions. As the results show, using the combination of PC-PMC and MLLR results in high system robustness in the presence of all noise types.

4 Summary and Conclusion

Nevisa, a Persian HMM-based, speaker-independent, continuous speech recognition system was introduced. Nevisa has been designed with the high degree of flexibility and modularity allowing a research framework into different aspects of the recognition and is suitable for various applications. It uses state of the art statistical and grammatical language models and is equipped with various robustness techniques. The current version of this engine is under further developments. Modeling context dependent acoustic phone units (e.g., tri-phones) and improving the lexicon and language models are the future improvements.

References

1. Rabiner, L.: Challenges in Speech Recognition and Natural Language Processing. In: SPECOM 2006, June 25 (2006)
2. Furui, S.: 50 Years of Progress in Speech and Speaker Recognition Research. ECTI Transaction on Computer and Information Technology 1(2) (November 2005)
3. Huang, X.D., Acero, A., Hon, H.: Spoken language processing. Prentice Hall, Englewood Cliffs (2000)
4. Rabiner, L., Juang, B.H.: Fundamentals of Speech Recognition. Prentice Hall, Englewood Cliffs (1993)
5. Rabiner, L.: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proceedings IEEE 77(2), 257–285 (1989)
6. Allen, J.: Natural Language Understanding. The Benjamin-Cummings Publishing Company, Inc. (1995)
7. Babaali, B., Sameti, H.: The Sharif Speaker Independent Large Vocabulary Speech Recognition System. In: The 2nd Workshop on IT & Its Disciplines, Kish Island, Iran (2004)
8. Sameti, H., Movasagh, H., Babaali, B., Bahrani, M., Hosseinzadeh, K., Fazel Dehkordi, A., Abutalebi, H.R., Veisi, H., Mokri, Y., Montazeri, N., Nezami Ranjbar, M.: Large Vocabulary Persian Speech Recognition System. In: 1st workshop on Persian language and computer, Tehran, Iran, May 25-26 (2004)

9. Movasagh, H.: Design and Implementation of an Optimized Search Method for HMM-based Persian Continuous Speech Recognition. M.S. thesis, Computer Engineering Department, Sharif University of Technology, Tehran-Iran (2004)
10. Babaali, B.: Incorporating Pruning techniques for improving the performance of an HMM-Based Continuous Speech Recognizer. M.S. thesis, Computer Engineering Department, Sharif University of Technology, Tehran-Iran (2004)
11. Ortmanns, S., Eiden, A., Ney, H.: Improved Lexical Tree Search for Large Vocabulary Speech Recognition. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Proc. (1998)
12. Haeb-Umbach, R., Ney, H.: Improvements in Time-Synchronous Beam Search for 10,000-Word Continuous Speech Recognition. IEEE Trans. on Speech and Audio Proc. 2, 353–356 (1994)
13. Bahrani, M., Sameti, H., Hafezi, N., Movasagh, H.: Building and Incorporating Language Models for Persian Continuous Speech Recognition Systems. In: Proc. 5th international conference on Language Resources and Evaluation, Genoa, Italy, pp. 101–104 (May 2006)
14. Bahrani, M., Sameti, H., Hafezi, N., Momtazi, S., Movasagh, H.: Using Persian Text Corps to Building Statistical Language Models for Persian Speech Recognition Systems. In: 2nd workshop on Persian language and computer, Tehran, Iran (2006)
15. Hafezi, M., Sameti, H., Mansuri, N., Montazeri, N., Bahrani, M., Movasagh, H.: A Grammatical Model for Improving the Performance of Persian Speech Recognition Systems. In: 2nd workshop on Persian language and computer, Tehran, Iran (2006)
16. Veisi, H.: Model-based methods for noise robust speech recognition systems, M.S thesis, Computer Engineering Department, Sharif University of Technology (November 2005)
17. Veisi, H., Sameti, H., Babaali, B., Hosseinzadeh, K., Manzuri, M.: Improving the Robustness of Persian Large Vocabulary Continuous Speech Recognition System for Real Applications. In: IEEE International Conference on Information & Communication Technologies, from Theory to Applications (ICTTA) (April 2006)
18. Hosseinzadeh, K.: Improving the Accuracy of Continuous Speech Recognition in Noisy Environments. M.S thesis, Computer department, Sharif University of Technology (November 2004)
19. Hosseinzadeh, K., Sameti, H., Abutalebi, H.R., Fazel Dehkordi, A.: MLLR method for environmental adaptation in the continuous FARSI speech recognition. In: The 6th Conference on Intelligent Systems, Kerman, Iran (2004)
20. Bijankhan, M., et al.: FARSDAT–The Speech Database of Farsi Spoken Language. In: Proc. The Fifth Australian Int. Conf. on Speech Science and Tech., perth, vol. 2 (1994)

Effects of Feature Domain Normalizations on Text Independent Speaker Verification Using Sorted Adapted Gaussian Mixture Models

Rahim Saeidi[1], Hamid Reza Sadegh Mohammadi[1],
Todor Ganchev[2], and Robert D. Rodman[3]

[1] Iranian Research Institute for Electrical Engineering, Tehran, I. R. Iran
[2] Wire Communications Laboratory, University of Patras Rio-Patras, 26500, Greece
[3]Department of Computer Science, North Carolina State University, NC 27695-8206, US
r.saeidi@ijece.org, h.sadegh@ijece.org,
tganchev@wcl.ee.upatras.gr, rodman@ncsu.edu

Abstract. In this paper we evaluate sorted Gaussian Mixture Model (GMM) system performance for Text Independent Speaker Verification under the feature domain normalization conditions. Sorted GMM is a speed-up algorithm proposed for GMM based systems. Cepstral Mean Subtraction (CMS) and Dynamic Range Normalization (DRN) are the normalization schemes studied for sorted GMM system purposes. Effectiveness of these normalizations has been proved in speaker recognition systems while their effectiveness on the speed-up of GMM based speaker verification is showed in this study. The baseline system is a universal background model–Gaussian mixture model (UBM-GMM) system and evaluations were performed on the NIST 2002 speaker recognition evaluation database with NIST SRE rules. It is shown that CMS and DRN normalizations enhance both the baseline system and sorted GMM system performances. In other words, the performance loss due to reducing the computational load is mitigated by applying CMS and DRN.

Keywords: Text independent speaker verification, speed-up, UBM, CMS, DRN, sorted Gaussian mixture model, NIST.

1 Introduction

Gaussian mixture model (GMM) is a common baseline system in speaker recognition applications [1]. Such a system is normally used to measure the effectiveness of novel algorithms in modeling approaches. GMM is a statistical approach for text independent speaker recognition with a high computational load during the test phase. A popular method for speaker verification is to model the speakers with GMMs based on the maximum-likelihood (ML) criterion, which has been shown to outperform several other existing techniques. In the state-of-the-art systems, speaker-dependent GMMs are derived from a speaker-independent universal background model (UBM) by adapting the UBM components with maximum a posteriori (MAP) adaptation using speakers' personal training data [2]. This method includes a natural association between the UBM and the speaker models; for each UBM Gaussian component, there is

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 493–500, 2008.

a corresponding adapted component in the speaker's GMM for each speaker. In the verification phase, each test vector is scored against all UBM Gaussian components, and a small number (typically 5) of the best scoring components in the corresponding speaker-dependent adapted GMM are scored. The decision score is computed as the Log Likelihood Ratio (LLR) of the speaker GMM and UBM scores. This procedure effectively reduces the computational complexity. In this study, this speaker verification system is used as the baseline to compare with the sorted GMM algorithm.

Chan *et al.* have categorized a four layer scheme for fast GMM computations, i.e., frame-layer, GMM-layer, Gaussian-layer, and component-layer [3]. We considered the Gaussian-layer scheme of this categorization as a reference point to describe the sorted GMM algorithm [4]. The speed-up concept of a GMM-UBM based system with pre-processing was previously investigated for speaker verification systems in the authors' earlier works [5]-[7]. Also, several approaches were reported as model domain speed-up schemes to reduce the computational complexity, such as the use of Gaussian selection, tree-structured Gaussian densities, hash GMM, VQ pre-classifier, structural GMM, and pruning methods [8]-[13], all of which degrade the system performance but marginally.

Effectiveness of the sorted GMM algorithm has been demonstrated in our previous works on a TV recorded Farsi database with 80 speakers. In this study, we evaluated the sorted GMM algorithm on NIST SRE 2002 database and showed its efficiency on spontaneous speech with different search width values as well as against multiple feature domain normalization methods. While there exist more efficient algorithms such as RASTA filtering [14], feature warping [15], short-time Gaussianization [16] and feature mapping [17] for feature normalization – which were evaluated on GMM based speaker verification systems [18], [19] – only two common types of feature domain normalization, namely cepstral mean subtraction (CMS) and dynamic range normalization (DRN), were used in this study. The three developed systems considered in this paper are: GMM-UBM without any normalization, GMM-UBM by applying CMS normalization, and GMM-UBM by applying CMS normalization and DRN.

The remainder of the paper is organized as follows. In Section 2, the sorted GMM method is described in detail. Section 3 presents the computer simulation and experimental results. Finally, Section 4 concludes the paper.

2 Sorted GMM Principles

The sorted Gaussian mixture model is a recently reported method for the fast scoring GMM that is outlined here. Given an L -dimensional feature vector $\mathbf{x}_t=[x_{1t},x_{2t},...,x_{Lt}]^T$ related to the speech frame at the time interval t, and a GMM of order M , we define a sorting parameter $s_t=f(x_{1t},x_{2t},...,x_{Lt})$, which is a scalar by definition, where $f(\cdot)$ is a suitable function known as a sorting function, chosen in such a way that neighboring target feature vectors provide almost neighboring values of s_t . Then the mixtures of the GMM are sorted in ascending order of the associated sorting parameter, according to the vector $\mathbf{S}=[s_1,s_2,...,s_M]^T$ with $s_1\le s_2\le...\le s_M$. In this research we simply considered $f(\cdot)$ as the sum of input vector elements and GMM mixtures were sorted in ascending order of the summation of their mean value components.

To compute the likelihood of the input feature vector, in the first step the quantity s_t is scalar quantized by $\mathbf{S}$. Suppose s_i is the result of scalar quantization, with $1 \le i \le M$. The index of s_i (i.e., i) is called the central index. In the next step, the input feature vector's likelihood is evaluated using the ordinary method by an extensive local search in the neighborhood of the central index, which includes a subset of M_s mixtures out of the entire mixtures $M_s < M$. For example, only the mixtures with indices within the range of $i-k+1$ to $i+k$ may be searched, where k is an offset value ($k = M_s / 2$). M_s is called the search width.

To achieve better performance for the sorted GMM, we always search $2k$ mixtures. For example, this means that for the case of $i \le k$, the first $2k$ mixtures in the GMM are considered for local search, and for $i \ge M - k$ the last $2k$ mixtures are evaluated for the likelihood calculation. Generally, the computational complexity of this method grows linearly with M_s, which normally is set to be less than M .This sorted GMM method can be applied to any GMM, such as a UBM, without any further training process. However, the performance can be further enhanced if the following optimization algorithm is used to optimize the GMM:

Step 1. *Initialization*: Set $n = 0$, $M_n = M_i$, where M_i is the initial GMM. Calculate the sorting parameter related to each mixture and sort the GMM in ascending order of the sorting parameter.

Step 2. *Likelihood Estimation*: Calculate the likelihood of the entire training database with M_n mixtures using the sorted GMM method.

Step 3. *GMM Adaptation*: Compute the M_{n+1} GMM. This is done simply by adapting each mixture from M_n using associated training vectors found in the previous Step.

Step 4. *Sorting*: Recalculate the sorting parameters related to the mixtures of the new GMM, M_{n+1}, and sort the GMM in ascending order of the sorting parameter. Also, set $n = n+1$.

Step 5. *Termination*: If the total likelihood is higher than a certain threshold (or any other reasonable condition), then stop the algorithm; otherwise go to Step 2.

After the optimization stage with the optimized UBM in hand, the speakers' GMMs are simply adapted from the optimized UBM like what is done in the ordinary GMM-UBM training. The memory storage required for the sorted GMM is $(2d+2)/(2d+1)$ times that needed for the ordinary GMM where d is the feature vector length (the negligible extra storage is required to store the sorting parameter quantization table). On the other hand, the number of Gaussian computations (which is used as a measure of speed-up) is reduced to M_S+C (where C is the number of top scoring mixtures whose corresponding mixtures are evaluated in the speaker GMM) which is less than $M+C$ Gaussian computations in the baseline system. Thus the speed-up factor of the sorted GMM algorithm is approximately equal to $(M+C)/(M_S+C)$.

3 Performance Evaluation

To evaluate the performance of the proposed method several experiments were performed. This section explains different aspects of these trials.

3.1 Speech Database

The speaker recognition experiments were conducted on cellular telephone conversational speech from the Switchboard corpus. This data was used by NIST for the 2002 one-speaker detection task [19]. Given a speech segment of about 30 seconds, the goal is to decide whether this segment was spoken by a specific target speaker or not. For each of 330 target speakers (139 males and 191 females), two minutes of untranscribed, concatenated speech is available for training the target model. Overall 3570 test segments (1442 males and 2128 females), mainly lasting between 15 and 45 seconds, have to be scored against roughly 10 gender-matching impostors and against the true speaker. The gender of the target speaker is known. We made use of a subset of the speech data from the NIST 2000 evaluation in order to train the universal background model. This data includes files from 200 development speakers with approximately 6.5 hours of speech (2 minutes of speech for each of 100 males and 100 female speakers) which are used to train the background model.

3.2 Evaluation Measure

The evaluation of the speaker verification system is based on detection error tradeoff (DET) curves, which show the tradeoffs between false alarm (FA) and false rejection (FR) errors. We also used the detection cost function (DCF) defined in [19]

$$DCF = C_{miss}.E_{miss}.P_{t\arg et} + C_{fa}.E_{fa}(1-P_{t\arg et}) \quad (1)$$

where $P_{t\arg et}$ is the *a priori* probability of the target tests with $P_{t\arg et}=0.01$ and the specific cost factors $C_{miss}=10$ and $C_{fa}=1$, so the point of interest is shifted towards low FA rates. Also, an equal error rate (EER) criterion was used as a measure of system performance which shows the point where FA and FR errors are equal.

3.3 Experimental Setup

We organized three sets of experiments to study the effects of feature domain normalization on sorted GMM system performance. The GMM-UBM system was used as a baseline in the benchmark. Three systems were trained and mutually compared: The GMM-UBM system with no normalization, with CMS normalization, and with CMS+DRN normalization. Moreover, their sorted GMM variants were also trained and evaluated. All UBM and speakers' model trainings were performed separately for each of these systems because it was believed that CMS and DRN normalizations may avoid some local convergence with respect to the EM algorithm.

The speaker recognition system applied a 16 millisecond Hamming window and an 8 millisecond frame rate to the speech signal to obtain a sequence of frames, and

calculate a feature vector for each frame. The feature vector consisted of 19-dimensional Mel-Frequency Cepstral Coefficients (MFCCs), which are a type of smoothed spectral representation, to which 19 delta MFCC coefficients were appended to form a 38-dimensional feature vector per frame. The zero cepstral coefficients (energy terms) were excluded in this study. It is common to process the MFCCs using CMS to reduce the effects of convolutional distortions. In this method, cepstral coefficients are averaged over the duration of an entire utterance, and the averaged values are subtracted from the cepstral coefficients of each frame. This method can compensate fairly well for additive variation in the log spectral domain. However, it unavoidably removes some text-dependent and speaker-specific features, so it is inappropriate for short utterances in speaker recognition applications. It has also been shown that time derivatives of cepstral coefficients (delta-cepstral coefficients) are resistant to linear channel mismatches between training and testing.

When additive noise exists, a natural extension of CMN is DRN, which normalizes the distribution of cepstral features over some specific window length by subtracting the mean and scaling the standard deviation. DRN yields a better compensation of the mismatch caused by additive noise. The block diagram of the system with MFCC extraction and DRN normalization is depicted in Fig. 1.

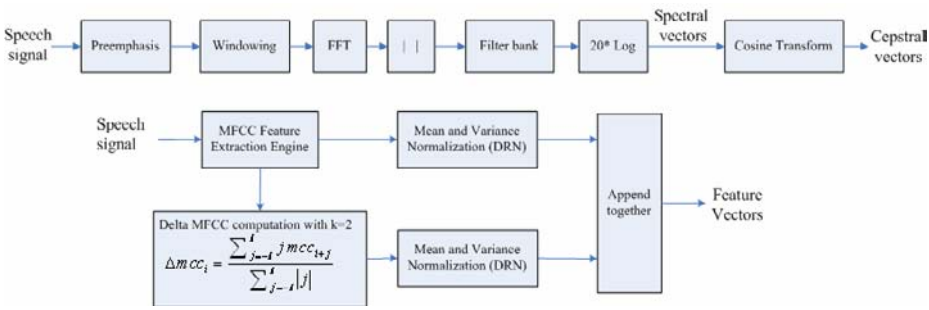


Fig. 1. System block diagram with MFCC extraction and dynamic range normalization modules

For each target speaker a speaker-specifi GMM with diagonal covariance matrices was trained via maximum *a posteriori* (MAP) adaptation of the Gaussian means of the UBM. The relevance factor, which is a parameter to determine the impact of new data in model adaptation from the UBM, was set to 16. (In this case the UBM model included 1024 Gaussians.) This model was trained on a total of about 6.5 hours of data from the 200 development speakers with at least 100 EM algorithm iterations. Variance flooring of 0.01 was imposed to avoid singularity.

For each verification test, i.e. a pair of a test segment and a target speaker, the test segment is scored against both the target model and the background model matching the target gender, ignoring low energy frames (Silence removal was performed by applying a bi-Gaussian modeling of frames energy and discarding frames with low mean values).

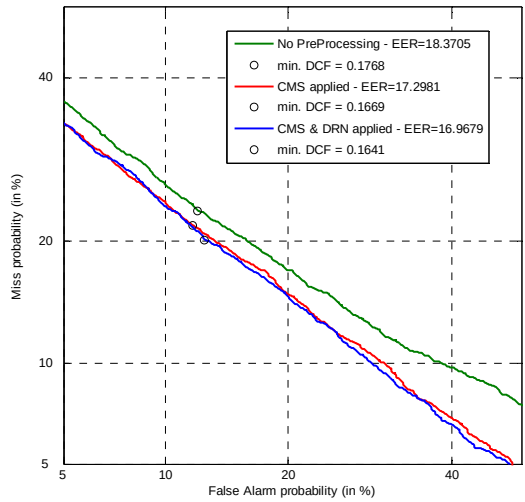


Fig. 2. Baseline system performance (without any speed-up on Gaussian computations) against two feature domain normalization methods

For a given test segment X and a target model λ_{target}, the decision score $S(X, \lambda_{target})$ is a log-likelihood ratio: $S(X,\lambda_{target}) = \log P(X|\lambda_{target}) - \log P(X|\lambda_{UBM})$. The number of top scoring mixtures C (recognized as top-C) was set to 5. For each verification test, i.e. a pair of a test segment and a target speaker, the test segment is scored against both the target model and the background model matching the target gender, ignoring low energy frames (Silence removal was performed by applying a bi-Gaussian modeling of frames energy and discarding frames with low mean values). For a given test segment X and a target model λ_{target}, the decision score $S(X, \lambda_{target})$ is a log-likelihood ratio: $S(X,\lambda_{target}) = \log P(X|\lambda_{target}) - \log P(X|\lambda_{UBM})$. The number of top scoring mixtures C (recognized as top-C) was set to 5.

It is noteworthy that we did not used any type of score domain normalizations, such as Znorm, Hnorm, Tnorm, etc. Search of the sorted GMM algorithm varied from 8 to 512 and the results were evaluated.

3.4 Experiments Results

The first experiment on the baseline system was conducted by applying the two previously described normalizations CMS and CMS+DRN. As Fig. 2 shows, the use of these normalizations has a considerable impact on the system performance in terms of EER and min. DCF values improvement. The extra gain achieved by using DRN in addition to CMS is marginal. This effect becomes more pronounced when we look at the results of the sorted GMM system performance where the use of DRN not only improves the performance of the GMM-UBM system but also improves the efficiency of the sorted GMM algorithm.

The second experiment was conducted to study the effect of channel normalization schemes on the sorted GMM algorithm performance. In this way we evaluated the

Table 1. Sorted GMM algorithm performance comparison on different normalization conditions for different search widths

Verification system	*EER*	*Min.DCF*	*Speed-up*
Baseline	18.3705%	0.1768	1
Sorted GMM – Search width = 512	18.6729%	0.1790	1.99
Sorted GMM – Search width = 256	19.1418%	0.1840	3.94
Sorted GMM – Search width = 128	19.6371%	0.1920	7.74
Sorted GMM – Search width = 64	20.7844%	0.2047	14.91
Sorted GMM – Search width = 32	25.5040%	0.2504	27.81
Sorted GMM – Search width = 16	31.5781%	0.3056	49
Sorted GMM – Search width = 8	37.9540%	0.3821	79.15
Baseline (+CMS)	17.2981%	0.1669	1
Sorted GMM (+CMS) – Search width = 512	17.3355%	0.1687	1.99
Sorted GMM (+CMS) – Search width = 256	17.5326%	0.1697	3.94
Sorted GMM (+CMS) – Search width = 128	18.0708%	0.1743	7.74
Sorted GMM (+CMS) – Search width = 64	19.1044%	0.1864	14.91
Sorted GMM (+CMS) – Search width = 32	21.9622%	0.2138	27.81
Sorted GMM (+CMS) – Search width = 16	27.0202%	0.2657	49
Sorted GMM (+CMS) – Search width = 8	33.7861%	0.3338	79.15
Baseline (+CMS & DRN)	16.9679%	0.1641	1
Sorted GMM (+CMS & DRN) – Search width = 512	16.9957%	0.1642	1.99
Sorted GMM (+CMS & DRN) – Search width = 256	17.0942%	0.1639	3.94
Sorted GMM (+CMS & DRN) – Search width = 128	17.4910%	0.1689	7.74
Sorted GMM (+CMS & DRN) – Search width = 64	18.8741%	0.1846	14.91
Sorted GMM (+CMS & DRN) – Search width = 32	21.4599%	0.2091	27.81
Sorted GMM (+CMS & DRN) – Search width = 16	25.9770%	0.2589	49
Sorted GMM (+CMS & DRN) – Search width = 8	32.4446%	0.3223	79.15

system performance by applying the sorted GMM as a speed-up engine for the baseline system. Results are presented in Table 1.

Table 1 shows the sorted GMM performance and the gain achieved by applying feature normalizations. Results show that rate of EER increase in consequence of system speed-up (by applying sorted GMM) decreased significantly when channel normalization schemes applied. Search width of 128 could be marked as the knee of sorted GMM under the experiments condition, because the performance falling down becomes terrible after this point in terms of EER and min. DCF.

4 Conclusion

In this paper we evaluated a sorted GMM with a fast scoring capability on a NIST database. The faster algorithm had a marginally higher memory requirement than an ordinary GMM. We also examined the effects of two channel normalization schemes. We showed that the CMS and DRN normalizations improve the performance of the sorted GMM algorithm.

References

1. Reynolds, D.A., Rose, R.C.: Robust text-independent speaker identification using Gaussian mixture speaker models. IEEE Trans. on Speech Audio Processing 3(1), 72–83 (1995)
2. Reynolds, D.A., Quatieri, T.F., Dunn, R.B.: Speaker verification using adapted Gaussian mixture models. Digital Signal Processing 10(1-3), 19–41 (2000)
3. Chan, A., Mosur, R., Rudnicky, A., Sherwani, J.: Four-layer categorization scheme of fast GMM computation techniques in large vocabulary continuous speech recognition systems. In: Proc. INTERSPEECH-2004, pp. 689–692 (2004)
4. Sadegh Mohammadi, H.R., Saeidi, R.: Efficient implementation of GMM based speaker verification using sorted Gaussian mixture model. In: Proc. EUSIPCO 2006, Florence, Italy, September 4-8 (2006)
5. Saeidi, R., Sadegh Mohammadi, H.R., Rodman, R.D., Kinnunen, T.: A new segmentation algorithm combined with transient frames power for text independent speaker verification. In: Proc. ICASSP 2007, vol. 1, pp. 305–308 (April 2007)
6. Sadegh Mohammadi, H.R., Saeidi, R., Rohani, M.R., Rodman, R.D.: Combined inter-frame and intra-frame fast scoring methods for Efficient implementation of GMM-based speaker verification systems. In: Proc. ICASSP 2007, pp. 309–312 (April 2007)
7. Auckenthaler, R., Mason, J.: Gaussian selection applied to text-independent speaker verification. In: Proc. A Speaker Odyssey, Speaker Recognition Workshop (2001)
8. Roch, M.: Gaussian-selection-based non-optimal search for speaker identification. Speech Communication 48, 85–95 (2006)
9. Xiang, B., Berger, T.: Efficient text-independent speaker verification with structural Gaussian mixture models and neural networks. IEEE Trans. Speech Audio Processing 11(5), 447–456 (2003)
10. Xiong, Z., Zheng, T.F., Song, Z., Soong, F., Wu, W.: A tree-based kernel selection approach to efficient Gaussian mixture model-universal background model based speaker identification. Speech Communication 48, 1273–1282 (2006)
11. Pellom, B.L., Hansen, J.H.L.: An efficient scoring algorithm for Gaussian mixture model based speaker identification. IEEE Signal Processing Lett. 5(11), 281–284 (1998)
12. Kinnunen, T., Karpov, E., Fränti, P.: Real-time speaker identification and verification. IEEE Trans. on Audio, Speech and Language Processing 14(1), 277–288 (2006)
13. Hermansky, H., Morgan, N.: RASTA processing of speech. IEEE Trans. on Speech and Audio Processing 2, 578–589 (1994)
14. Pelecanos, J., Sridharan, S.: Feature warping for robust speaker verification. In: Proc. ISCA Workshop on Speaker Recognition - 2001: A Speaker Odyssey (June 2001)
15. Xiang, B., Chaudhari, U., Navrátil, J., Ramaswamy, G., Gopinath, R.: Short-time Gaussianization for robust speaker verification. In: Proc. ICASSP, vol. 1, pp. 681–684 (2002)
16. Reynolds, D.A.: Channel robust speaker verification via feature mapping. In: Proc. ICASSP, vol. II, pp. 53–56 (April 2003)
17. Barras, C., Gauvain, J.L.: Feature and score normalization for speaker verification of cellular data. In: Proc. ICASSP, vol. 2, pp. 49–52 (2003)
18. Burget, L., et al.: Analysis of feature extraction and channel compensation in GMM speaker recognition system. IEEE Transactions on Audio, Speech, and Language Processing 15(7), 1979–1986 (2007)
19. The NIST year 2002 speaker recognition evaluation, `http://www.nist.gov/speech/tests/`

A Centrally Managed Dynamic Spectrum Management Algorithm for Digital Subscriber Line Systems

Adnan Rashdi[1], Noor Muhammad Sheikh[2], and Asrar ul Haq Sheikh[3]

[1] Research Scholar, Research Center, University of Engineering and Technology, Lahore 54890, Pakistan
[2] Dean of Faculty, Electrical Engineering Department, University of Engineering and Technology, Lahore 54890, Pakistan
[3] Bugshan/Bell Labs Chair Professor, Electrical Engineering Department, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia
adnanrashdi@ieee.org, deanee@uet.edu.pk, asrarhaq@kfupm.edu.sa

Abstract. This paper presents a centrally managed scheme of combined crosstalk cancellation and multiuser power control for minimizing crosstalk degradation effects in multiuser digital subscriber line (DSL) systems. Exploiting the frequency selective nature of the heavily unbalanced crosstalk channel, the crosstalk cancellation is carried out on the selected tones in order to reduce complexity. Multiuser power control is implemented centrally through Spectrum Management Center (SMC) that knows the channel gains and noise power spectral density (PSD) for all users. Simulation results show that the proposed algorithm considerably enlarges the data rate region when applied to very high speed digital subscriber lines (VDSL) upstream transmission.

Keywords: Digital subscriber line (DSL), multiuser power allocation and control, waterfilling, multiuser detection and crosstalk cancellation.

1 Introduction

Crosstalk is a dominant source of performance degradation and significantly limits the data rate and reach at which a DSL service may be provided [1]. Near-end crosstalk (NEXT) can be avoided using frequency division duplexing (FDD) but far- end crosstalk (FEXT) is still a major problem in most DSL systems. This is particularly true when one of the transmitters is located much close to the receiving modems than all other transmitters. The crosstalk from this transmitter can often be stronger than the signals of interest on the other lines, leading to a total loss of service. This so-called *near-far* problem is particularly evident in upstream VDSL transmission when customer premises (CP) modem is located further upstream of the other modems in the network [2].

Current research is focusing on dynamic spectrum management (DSM) techniques to minimize crosstalk effects in multiuser communication environment. DSM facilitates coordination among the mutually interfering lines in a binder at spectral and signal level. With spectral coordination at DSM Level-2, multiuser power allocation and control minimizes the negative effects of the crosstalk by varying the transmit

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 501–508, 2008.

power spectral density (PSD) of the modems within a network. Signal coordination at DSM Level-3 enables multiuser detection (crosstalk cancellation). Full crosstalk cancellation increases run time complexity of the DSL modems significantly as compared to multiuser power allocation and control. Therefore the previous studies independently considered multiuser power allocation and control [3] [4] and crosstalk cancellation [5] [6] [7] based techniques to mitigate crosstalk degrading effects in different multiuser DSL networks.

This paper proposes a centrally managed scheme of combined crosstalk cancellation and multiuser power control which is applicable to most of the DSL systems suffering from near-far problem. Crosstalk cancellation is carried out at the tones with strong crosstalk followed by power allocation and control based on multiuser bit loading [8] to achieve a stable PSD. We have simulated multiuser VDSL environment in MATLAB and the proposed algorithm has been tested for near-far scenario considering upstream transmission. Simulation results show that the proposed algorithm performs better than distributed multiuser power allocation and control algorithm (iterative waterfilling) [3][4] and gives an enlarged data rate region.

2 System Model and Problem Formulation

Most current DSL systems use Discrete Multi-Tone (DMT) modulation. The DMT divides the frequency selective channel in a number of parallel subchannels or tones. Each tone is capable of transmitting data independently from other tones and so the transmit power and the number of bits can be assigned individually for each tone. Transmission for a binder of n users (n = 1…. N) can be modeled on each tone k by

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{z}_k \quad k = 1.... K \tag{1}$$

The vector $\mathbf{x}_k = [x_k^{\,1}, x_k^{\,2}, ..., x_k^{\,N}]$ contains the transmitted signals on tone k for all users N. $x_k^{\,n}$ is the signal transmitted onto line n at tone k. The transmit PSD of user n on tone k is denoted by $s_k^{\,n} = \varepsilon\left\{\left|x_k^{\,n}\right|^2\right\}$. $\mathbf{y}_k = [y_k^{\,1}, y_k^{\,2}, ..., y_k^{\,N}]$ contains the received signals on tone k for all users N. $\mathbf{z}_k$ is the vector of additive white Gaussian noise (AWGN) on tone k for all users N. $z_k^{\,n}$ is the noise experienced by user n on tone k with PSD $\sigma_k^{\,n} = \varepsilon\left\{\left|z_k^{\,n}\right|^2\right\}$. $\mathbf{H}_k$ is the N x N channel transfer matrix on tone k. $h_k^{n,m} \triangleq [\mathbf{H}_k]_{n,m}$ is the channel from user m to user n on tone k. The diagonal elements of $\mathbf{H}_k$ contain the direct channels while the off diagonal elements contain the crosstalk channels. It is assumed that all receiving modems are colocated and synchronized. We denote symbol period as T and frequency width of the DMT tones as Δf. Assuming that all the transmitted signals and background noises are Gaussian, the achievable bit loading of user n on tone k is

$$b_k^n \triangleq \log_2\left(1+\frac{1}{\Gamma}\cdot\frac{\left|h_k^{n,n}\right|^2 s_k^n}{\sum_{m\neq n}\left|h_k^{n,m}\right|^2 s_k^m+\sigma_k^n}\right) \tag{2}$$

where Γ is the SINR gap to capacity which is the function of the desired bit error rate (BER), the noise margin, and the coding gain. The data rate of user *n* is then

$$R^n=\frac{1}{T}\sum_{k=1}^{K} b_k^n \tag{3}$$

Our interest is to maximize the achievable data rate region (union of all the data rate sets) for the best tradeoff of data rates among near and far users. Thus we define the problem as:

$$\text{maximize} \quad \{(R^1,R^2,.....,R^N): P^n \le P^{\max,n} \text{ for } n=1,.....,N\} \tag{4}$$

where $P^n=\Delta f\sum_{k=1}^{K} s_k^n$ is the power of user *n* with $s_k^n \ge 0 \forall n,k$ and $P^{\max,n}$ is the maximum power for user *n*. Defining rate sum maximization problem (RSMP) with a weighted power sum constraint:

$$\text{maximize} \sum_{n=1}^{N} R^n \tag{5}$$

$$\text{subject to} \sum_{n=1}^{N} w^n P^n \le P^{\max} \tag{6}$$

Equivalent to RSMP is weighted power sum minimization problem (WPSMP) which minimizes the weighted power sum for a given target rate sum:

$$\text{minimize} \sum_{n=1}^{N} w^n P^n \tag{7}$$

$$\text{subject to} \sum_{n=1}^{N} R^n \ge R^{target} \tag{8}$$

$P^{\max}$ is the maximum possible weighted power sum, w^n is the positive weight of user *n* and R^{target} is the target rate sum in (8). WPSMP is generalization of total power sum minimization problem discussed in [8] which minimizes the total power for a given target rate sum using multiuser discrete bit loading.

3 Centrally Managed Dynamic Spectrum Management Algorithm

Space (users) and frequency (tones) are two possible dimensions for the detection of strong crosstalk in a multiuser communication environment. The near-far effect in DSL systems gives rise to the space selectivity and facilitate in the detection of

strong crosstalker. The second dimension can be explored by selecting the tones where crosstalk cancellation can give the most benefit. Tone selection (TS) exploits the frequency selective nature of the crosstalk channel. In the low frequency bands, crosstalk coupling is small therefore the gain is achieved by the crosstalk cancellation becomes small in low frequency bands. On the other hand, the direct channel attenuation is so large in the high frequency bands that is hard to load many bits in the absence of the crosstalk. Therefore the data rate improvement due to the crosstalk cancellation is most noticeable in the intermediate frequencies. TS is a single dimension selective process which selects tones with strong crosstalk. Similar schemes have been proposed for wireless and wired systems in [9] [10]. With TS high computational complexity of full crosstalk cancellation can be reduced.

Our centrally managed DSM algorithm performs crosstalk cancellation based on TS followed by centralized multiuser power control based on multiuser discrete bit loading. Thus the proposed algorithm combines two approaches: crosstalk cancellation and crosstalk suppression to achieve enlarged data rate region for interference limited DSL channel. This methodology has never been considered in the past because unfortunately full crosstalk cancellation has a high run time complexity of $O(N)$ multiplications per tone per DMT-block per user, leading to a total complexity of $O(KN^2)$ where $O(.)$ is the order, K is number of tones and N is number of users. TS process leads to an implementable run time complexity of K_s multiplications per DMT-block per user where K_s is the number of tones selected for crosstalk cancellation out of K tones. Thus the proposed algorithm gives performance at the cost of very little run time complexity. Centrally managed dynamic spectrum management algorithm runs as follows:

3.1 Crosstalk Cancellation Based on Tone Selection (Stage-1)

Considering fixed transmit PSD s_k^n for user n on tone k; $b_k^n(full)$ is defined as the rate achieved by user n on tone k with full crosstalk cancellation

$$b_k^n(full) = \log_2\left(1+\frac{1}{\Gamma}\cdot\frac{\left|h_k^{n,n}\right|^2 s_k^n}{\sigma_k^n}\right) \tag{9}$$

$b_k^n(no)$ is defined as the rate achieved by user n on tone k with no crosstalk cancellation

$$b_k^n(no) = \log_2\left(1+\frac{1}{\Gamma}\cdot\frac{\left|h_k^{n,n}\right|^2 s_k^n}{\sum_{m\neq n}\left|h_k^{(n,m)}\right|^2 s_k^m+\sigma_k^n}\right) \tag{10}$$

Defining the gain of full crosstalk cancellation

$$g_k^n = b_k^n(full) - b_k^n(no) \tag{11}$$

and the indices of the tones ordered by this gain

$$\left\{k^n(1),.....k^n(K)\right\} \text{such that } g_k^n(i) \geq g_k^n(i+1), \forall i \tag{12}$$

After sorting the tones, the multiuser detector of interference cancellation type at the receiver of user n performs full crosstalk cancellation on the K_s important tones where gain is maximal and no crosstalk cancellation is done on all other tones.

3.2 Multiuser Power Control (Stage-2)

Initialization. Let the weight $w^n = w^{n,initial}$ for n=1,2,.....N

Multiuser Power Allocation. Allocate power over all users and tones using the following multiuser bit loading algorithm with given power weights.

Initialization. For all users and tones, calculate the cost to transmit one bit. Cost to increase one bit in tone k for user n is defined as the weighted incremental power sum:

$$J_k^n = \sum_{m=1}^{N} w^m \Delta s_k^{m,n} \tag{13}$$

Where $\Delta s_k^{m,n}$ is the incremental power of user m to add one bit to user n.

Bit-Loading Iterations. Repeat the following until a desired rate-sum is achieved; increase one bit in the user and tone pair (n,k) where adding one bit requires the minimum cost among all available users and tones, update cost to increase one bit of user n in tone k.

Rate Test: If the desired rate set is achieved, stop here.

Assigning New Weights: If the data rate of user n needs to be increased, reduce w^n by setting $w^n = w^n / \delta$. If the data rate of user n needs to be decreased, raise w^n by setting $w^n = w^n \delta$. Repeat the multiuser power allocation with the new weight until a desirable rate set is achieved.

The algorithm can be implemented with the support of a centralized agent or spectrum management center (SMC) placed at central office (CO) or optical network unit (ONU). The amount of power assigned to each user can be determined by the power weights in (13). If the weight w^n of user n is reduced, the cost function is less influenced by the power increase of user n, resulting in the faster allocation of the power of user n. Consequently, more power is assigned to user n with reduced weight w^n.

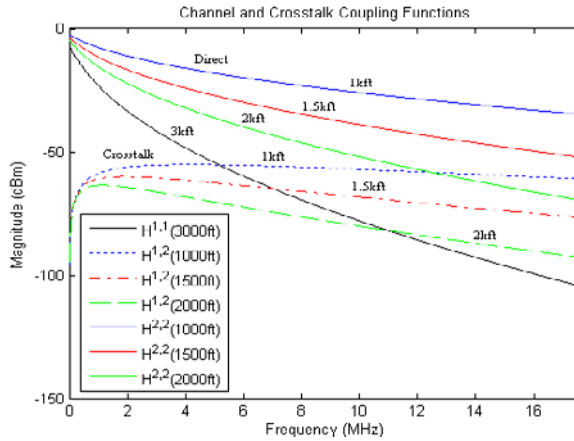


Fig. 1. Channel transfer functions and crosstalk coupling functions: Group 1 user at 3000ft and Group 2 user at 1000ft, 1500ft and 2000ft

On the other hand, less power is assigned to user n by raising the weight of user n. The amount of power assigned to each user is also controlled by the weights since the data rate is determined by the amount of power used. Usually for DSL channel $\delta = 8$ is used to change the data rate set.

4 Simulation Results

In this section, the performance of the proposed algorithm is shown by simulating VDSL systems using MATLAB. The simulation parameters are taken from ANSI VDSL standards [11]. The target symbol error probability is 10^{-7} or less. The coding gain and noise margin are set to 3dB and 6dB respectively. The maximum power is 14.5 dBm with the SNR gap of 12 dB. The number of bits in each tone are limited to be less than or equal to 15. FDD bandplan 998 (0.03-0.138 MHz band, 3.75-5.2 MHz band and 8.5-12 MHz band) has been used for upstream transmission. Near-far scenario with CO based DSL deployment has been simulated considering a cable binder having two groups. Group 1 has four users (1-4) located at 3000 ft from the central office and group 2 also has four users (5-8) located at varying loop length ranging from 1000 ft to 2000 ft from the central office. The channel transfer functions and the crosstalk coupling functions considering a group 1 user at a loop length of 3000 ft and a group 2 user at a varying loop length from 1000ft to 2000ft have been plotted in Fig.1 considering 0.4 mm (26 AWG) line. $H^{n,m}$ refers to the transfer function from user m to user n on all tones. The crosstalk transfer function is computed using the FEXT crosstalk model [11] where FEXT cross coupling increases with frequency as f^{2}. The crosstalk noise model A is used to generate crosstalk from other services. Crosstalk channels exhibit both space and frequency selectivity.

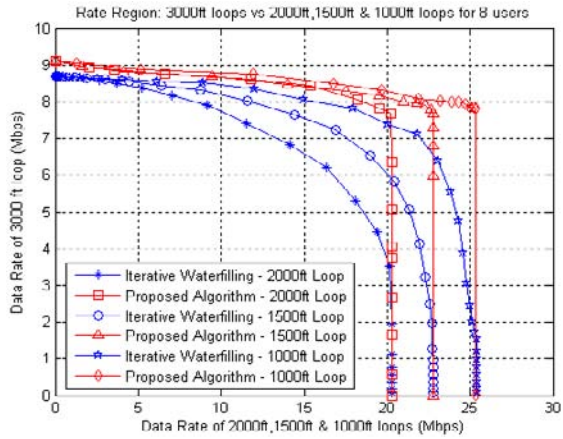


Fig. 2. Rate regions considering Group 1 user (1-4) at 3000ft and Group 2 user (5-8) at 2000ft, 1500ft and 1000ft loop length

The proposed algorithm is compared with distributed multiuser power and control algorithm (iterative waterfilling) [3,4]. Fig.2 shows the achievable data rate regions associated with the proposed algorithm and the distributed multiuser power allocation and control algorithm (iterative waterfilling) for 1000ft, 1500ft and 2000ft loop lengths. The proposed algorithm considerably enlarges the data rate region for these loop lengths as compared to iterative waterfilling. Since the data rates of users at the same distance from central office are almost the same, we took the average of data rates of the users with the same loop length to draw the rate regions in two dimensions.

5 Conclusions

We examined near far problem for DSL and proposed a combined crosstalk cancellation and power control scheme for DSL. Previous studies considered crosstalk as an additive noise while performing power allocation and control because of computational complexity of full crosstalk cancellation.

References

1. Cioffi, J.: DSL Advances. Prentice-Hall, Upper Saddle River (2002)
2. Chung, S.: DSL Handbook. Auerbach Publications, Boston (2004)
3. Yu, W., Ginis, G., Cioffi, J.: Distributed Multiuser Power Control for Digital Subscriber Lines. IEEE JSAC 20(5), 1105–1115 (2002)
4. Yu, W., Ginis, G., Cioffi, J.: An Adaptive Multiuser Power Control for VDSL. In: IEEE GLOBECOM, vol. 1, pp. 394–398. IEEE Press, New York (2001)
5. Ginis, G., Cioffi, J.: Vectored Transmission for Digital Subscriber Line Systems. IEEE JSAC 20(5), 1085–1104 (2002)

6. Zeng, C., Cioffi, J.: Crosstalk Cancellation in ADSL Systems. In: IEEE GLOBECOM, vol. 1, pp. 344–348. IEEE Press, New York (2001)
7. Cheong, K., Choi, W., Cioffi, J.: Multiuser Soft Interference Cancellation via Iterative Decoding for DSL Applications. IEEE JSAC 20(2), 363–371 (2002)
8. Lee, J., Sonalkar, R., Cioffi, J.: Multiuser Bit Loading For Multicarrier Systems. IEEE Trans. on Commun. 54(7), 1170–1174 (2006)
9. Gore, D., Paulraj, A.: Space Time Block Coding with Optimal Antenna Selection. In: International Conference on Acoustic, Speech and Signal Processing, pp. 2441–2444 (2001)
10. Cendrillon, R., Moonen, M., Ginis, G., Acker, K., Bosteoen, K., Vandaele, P.: Partial Crosstalk Cancellation exploiting line and tone selection in upstream DMT-VDSL. EURASIP Journal on Applied Signal Processing, 1520–1535 (2004)
11. Very-high-speed Digital Subscriber Lines (VDSL) Metallic Interface. T1E1.4/2003-210R1, Montreal, Canada, August 22-26

Methods for Analyzing Information Contained in an Enterprise Email Database

Mohsen Sadeghi[1], Khaled Hadj-Hamou[1], and Mickaël Gardoni[2]

[1] G-SCOP: Laboratoire des sciences pour la conception, l'optimisation et la production (INP Grenoble), 46, avenue Félix Viallet 38031 Grenoble France
[2] LGECO : Laboratoire du Génie de la Conception, (INSA Strasbourg) 24, Bd de la Victoire 67084 STRASBOURG, France
mohsen.sadeghi@g-scop.inpg.fr, khaled.hadj-hamou@g-scop.inpg.fr, mickael.gardoni@insa-strasbourg.fr

Abstract. Email is one of the most successful asynchronous communications yet devised. Many Researchers and Scientists often spend large proportions of their time using email for information and knowledge sharing. Research has not yet addressed how we can use emails as a source of information and knowledge. This study therefore presents a quantitative analysis of the emails to address these new questions. We discus the challenges that arise in email analyzing and classification We provide background, procedures for using natural language processing and text mining techniques for dealing with automatic knowledge extraction from email database.

Keywords: Email database, automatic indexing, classification, information retrieval, knowledge extraction.

1 Introduction

Email is the asynchronous communication method that has been the subject of many studies. These studies have focused on the communicative aspects of email without addressing how people can use and manage emails as a source of information and knowledge. This study therefore presents a quantitative analysis of emails to address these new questions by managing the whole knowledge cycle (such as identification, creation, reformulation, capitalization, sharing of knowledge).

Development of Email knowledge cycle has many complications that make it different from traditional methods. Some email messages only make sense in the context of previous messages therefore related messages should be searched and classified in the same class. The similarity of content associated with a certain email in different subjects could also cause the problem of content conflict in the email classification process. These complications together with lack of standards in classification methods have always been great obstacles for researches in this domain.

The analysis of email flows to and from a user's email account(s) reveals a tremendous amount of information about a person's interests, activities and behaviors that cannot be derived alone from content analyses of individual emails (stolfo, 2006). Based on recent works in text mining combined with analysis of a corpus of texts-Natural

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 509–517, 2008.

Language Processing (NLP)-, it may be possible to build a ontology of types of legal arguments and their relations, in much the same way as research that has been attempting to classify general argumentation schemes in exchanged emails (Feteris, 2000).

An email can be perceived as a text containing specific characteristics defining essential information such as the field which the email address of the recipient appears (To, CC, BCC) with date, subject, body and sometimes the replied body. Regarding these characteristics, the problem of automatic treatment of emails could be divided into three parts: collecting and treating emails of different users, content analysis and arrangement of messages and finally information retrieval and knowledge extraction.

This paper highlights an implementation of a rich combination of these ideas in order to make a practical, extensible system used for knowledge extraction from email database. The approach consists of two major part, the email analysis and modeling part and the evaluation and extraction part. The email analysis part converts semi structured data (emails) into structured data, stored inside the email database and applies the text mining technique to automatically classify the email with a high degree of accuracy. The second part extracts the essential information in these data.

Figure 1 shows the components explored in email mining process: indexing, classification, validation and extraction. The following subsections will describe each component in detail.

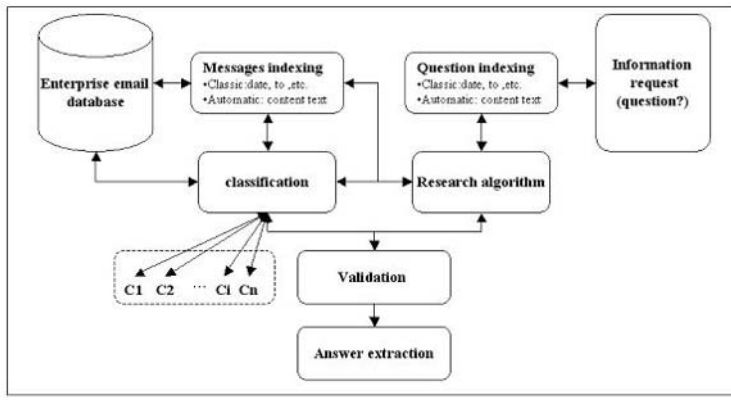


Fig. 1. Email mining process

2 Email Database

The first stage of email mining process is about how to structure emails in a database to facilitate the accessibility, compatibility and interoperability of their contents. We propose a UML database that can be used to define the email information elements.

Figure 2 shows different parts of information that would be stored in an email database such as subject, content, date, etc.

This model could be used as an indicator of collaboration and knowledge exchange between researchers. This is usually done by linking different emails communicated between different people on the same subject.

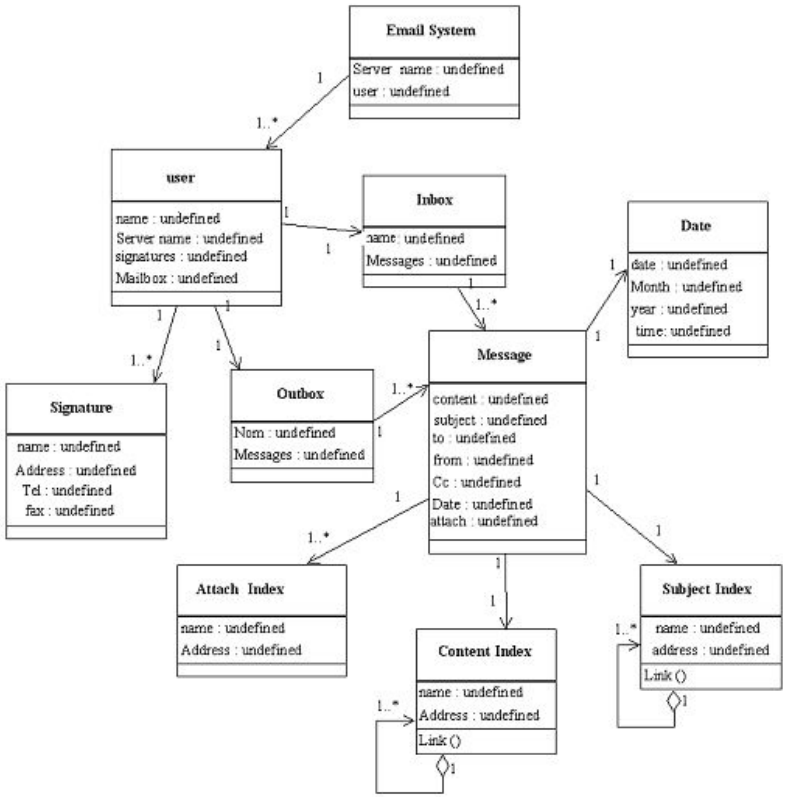


Fig. 2. Email database

Most emails include an address that identifies the sender. We can analysis the addresses from the "To:", "Cc:" or "Bcc:" fields of all the emails passed through the system to find a direct relation between them. The first step in is to automatically extract all such addresses to produce a list of < server-name, user-name; email > identifiers (IDs). Once this is done, we execute an algorithm that searches the similarity between IDs. This algorithm could find related emails if either the server-names are similar, or if the user-names are similar.

We use geometric and linguistic analysis to identify different elements of an email and to arrange each identified element in the database (Figure 3). This is useful in providing automatic analysis of the whole email content including: header, body, replied part, and attachments. This analysis includes:

Geometric and linguistic analysis to separate header, body and the replied part and to store them in a data base. For example: a reply text can generally be placed in any position in the e-mail document and each line is usually prefixed with a special character (e.g., ">").

Creating a link between message body part and message reply part if there exists.

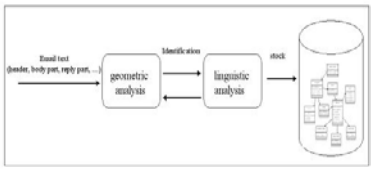


Fig. 3. Identification of email information elements

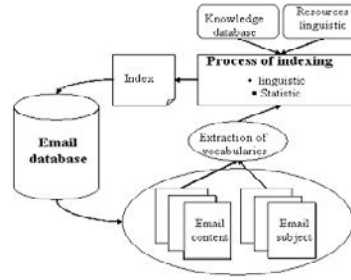


Fig. 4. Email indexing process

Skipping the analysis of attached files: They stored without being analyzed because it is not possible to identify the format of an attached file.

3 Email Indexing

Indexing texts using natural language processing is a method that is typically used with large text databases in order to decrease the information retrieval time during retrieval process (Arampatzis, 97). This process reads a document and automatically extracts words and phrases from unrestricted text and organizes them into an index table file that integrates syntactic, semantic, and morphological relationships. In addition, it uses an extensive system of knowledge-based morphological rules and functions to analyze words that are not already in its lexicon, in order to construct new lexical entries for previously unknown words (Moens, 2007). In addition to rules for handling derived and inflected forms of known words, the system includes rules for lexical compounds and rules that are capable of making reasonable guesses for totally unknown words. The creation of this index is external to files and does not affect them in any way.

Email indexing is a rich and interesting task. An indexing process allows search engine motors to retrieve relevant emails quickly and easily, thereby increasing efficiency. It differs from the standard text indexing in its highly subjective and specific text characteristic.

Emails are regularly being exchanged and even their major common topics are changing over time. The information content in an email includes simple text. Senders use natural language text with special electronic language such as “emoticons” (a sequence of ordinary characters you can find on your computer keyboard, ex.:-D laughing :) to transfer the information. Therefore, an email indexing system should be adaptive to the email text style.

We suggest the automatic indexing to represent the content and the subject of an email (figure 4) as follows:

Application of statistical or linguistical content analysis to provide email indexing creation of linguistic resources (dictionaries, morphological analyzer, syntactic analyzer, etc.) and knowledge database to establish the characteristic of different words

that have been used in specific domains by integrating user-specific resources creation of links between index files and original files in database

4 Classification

Classification of emails is a well known problem that addresses the applications and implications of computer algorithms to separate emails into predefined classes according to their contents. The result of classification can be seen as a representation of the content of a set of similar emails in order to facilitate the finding of related emails.

The latest research shows that considerable work has been done in the field of automatic classification of text documents through techniques such as Nearest-Neighbor Classification, naive Bayes, and decision trees. But, there has not yet been much work in how to classify emails automatically.

Although email classification can be viewed as a special case of text classification, the characteristics of emails text poses certain challenges, not often encountered in text or document classification. In this section we describe how text mining method can be used to analyze and classify emails with a high degree of accuracy (figure 5).

Our approach is based on the premise that for classifying incoming email messages content similarity identification can be used on pre-classified emails. This process compares a new message with the ones previously received, in order to find a similar class. The similarity between messages is established based on the comparison of message content with representation of each class (class index).

We use a class index where overlapping content is indexed among all emails that contain it. Emails are logically classified using a class representation index where related emails are organized as nodes in a graph (UML Email database). Thus, index space and index build times are greatly reduced. Similarity evaluation is faster due to the reduced index size.

The email content information such as the To, From and etc. play an important role in classifications. So, it is possible to distinguish two assumptions for classifying by the separation-modeling or the modeling-separation.

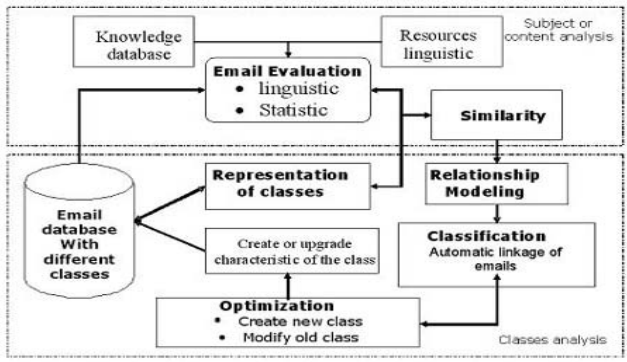


Fig. 5. Emails classification process

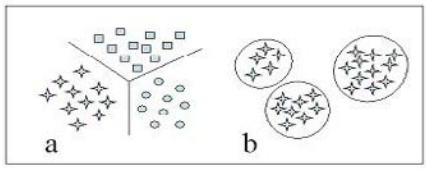


Fig. 6. Separation-Modeling, a) Separation b) Modeling

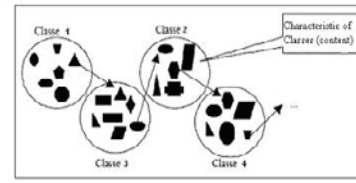


Fig. 7. Modeling-separation

Separation-Modeling (Figure 6). The model of Separation-Modeling consists of arranging the emails in sender-receiver groups. At the first stage, the groups of the emails are created according to each sender and receiver, then make the classification of the contents and the subject of the emails in each group and arrange them by chronological order. The objective is to establish content or subject relations between the emails of each sender and receiver.

Modeling-separation (Fig. 7). model of Modeling-separation consists of classification of all emails by content or subject for any sender and receiver and then make a chronological arrangement or sender-receiver arrangement for each class. Each class collects emails with similar contents (for example project). The principal goal of this classification is to have a summary representation of the contents in order to identify the emails of a certain subject or a certain context.

Certain emails present in a class can be also found in other classes. This makes it possible to build the link between two similar classes.

5 Extraction and Validation

Information extraction is typically performed to identify the emails and a set of information that may contain the answer to the query. it consists in automatically finding the relevant information from email database and find specific answers to: traditional query (request on date, user and etc.)
automatic query(request on contents)

The request on contents is based on a query representation in natural language and email classes to find the relevant content. The interactions between the stage of classification and the stage of extraction are illustrated by figure 8.

The Extraction and Validation system is a search engine produced by a linguistic and statistic analysis of the query and the email database. The system is composed of following modules:

Query analysis. The stage analyzes the query which is formulated in the natural language format or in the traditional query format. In content based query, this stage provides a representation of the query through the linguistic analyzer which will be used at the subsequent stages. The result is a query composed of a list of the linguistic elements extracted from the analysis (morphological analyzer, syntactic analyzer, etc.).

Candidate emails selection. In this stage, the system compares the query with the reference group files to find similarity between them. The goal of this stage is to restrict the number of emails in which the answer is sought.

Candidate emails analysis. It is necessary to carry out a more detailed analysis of the candidates emails with the algorithms known in Text Mining. The system uses a statistical technique. The goal of the statistical analysis is to be able to compare intersection between queries and the email index. It is used to give a ranked list of emails, according to their relevance. In addition, an extensive knowledge-based system is used to recognize the objects, the significant topics and the linguistic rules of the index fields in statistical analysis.

Extraction and creation of answer. In the final stage, the system retrieves the ranked, relevant emails from the indexes according to the corresponding formulated query and then merges the results obtained taking into account the original terms of the query (before formulation) and their weights in order to score the emails.

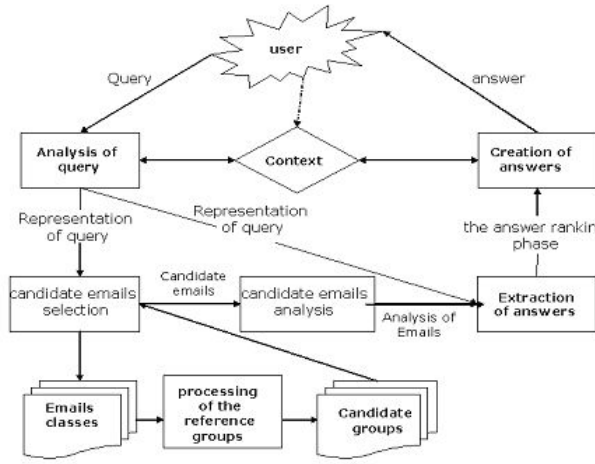


Fig. 8. Informatio n extraction and validation process

6 Conclusion

Email has become one of the most important methods of communication. The information in e-mail has the potential to add to the enterprise's knowledge assets. Today's E-mail softwares were not created to support knowledge management process. It has become necessary for organizations to consider email as part of the knowledge resources that require organization.

This paper provides a description of the process used to create a structure which supports the process of knowledge extraction in email database. We used the combination of statistical and linguistical approaches to increase the performance of information organization, structuring, retrieval and extraction. We explored four complementary processes: email storage, email indexing, email classification, and

information extraction and validation. Our experience with email classification was not very complete because the development of knowledge database and linguistic resource used in indexing our email storage still needs much work, especially for facilitating queries that exploit specific.

Acknowledgments

The authors wish to express their thanks to the sponsors of this study within the Bassetti socity for their interest and assistance. The research presented here benefited greatly from the contributions of other members of the Bassetti study team. Particular thanks also to Gregory poussier and Leyli Zafari for their advice, assistance, and support.

References

1. Ambroziak, J., Woods, A.: Natural language technology in precision content retrieval. In: Proceedings, Natural Language Processing and Industrial Applications (NLP+IA 1998), Moncton, New Brunswick, CA (1998)
2. Arampatzis, A., Tsoris, T.: IRENA: Information retrieval engine based on natural language analysis. In: Proceedings, Intelligent Multimedia Information Retrieval Systems and Management (RIAO 1997), Montreal, pp. 159–175 (1997)
3. Bird, C., Gourley, A., Devanbu, P.: Mining email social networks. In: Proceedings of the international workshop on mining software repositories, Shanghai, China (2006)
4. Brutlag, J.D., Meek, C.: Challenges of the Email Domain for Text Classification. In: Proceedings of the Seventeenth International Conference on Machine Learning (2000)
5. Carenini, G., Ng, R.T., Zhou, X.: Summarizing Email Conversations with Clue Words. In: Proceedings of the 16th international conference on World Wide Web, Banff, Alberta, Canada (2007)
6. Carenini, G., Ng, R., Zhou, X.: Discovery and regeneration of hidden emails. In: Proceedings of the 2005 ACM symposium on Applied computing, Santa Fe, New Mexico (2005)
7. Carolyn, J., Yang, B.: Experiments in automatic statistical thesaurus construction. In: Proceedings, 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1992), Copenhagen, pp. 77–88 (1992)
8. Cohen, W., Singer, Y.: Context-sensitive learning methods for text categorization. J. ACM 17(2), 141–173 (1999)
9. Coft, W.: Improving the effectiveness of information retrieval with local context analysis. ACM Transactions on Information Systems 18(1), 79–112 (2000)
10. Dasarathy, B.V.: Nearest-Neighbor Classification Techniques. IEEE Computer Society Press, Los Alamitos (1991)
11. Domingos, P., Pazzani, M.J.: On the optimality of the simple bayesian classifier under zeroone loss. Machine Learning 29(2/3), 103–130 (1997)
12. Feteris, E.T.: A dialogical theory of legal discussions: Pragma-dialectical analysis and evaluation of legal argumentation. Artif. Intell. Law 8(2/3), 115–135 (2000)
13. Feledman, R., Dagan, I.: KDT-Knowledge Discovery in Texts. In: Proceeding of the Conf. on Knowledge Discovery (KDD) (1995)
14. Gachot, I.: Linguistique + statistique + informatique = indexation automatique? Archimag 84, 34–37 (1995)

15. Karlsson, F.: Constraint Grammar as a framework for parsing running text. In: Proceedings, 13th International Conference on Computational Linguistics (COLING 1990), Helsinki, pp. 68–173 (1990)
16. Mani, I., Mark, T.: Advances in Automatic Text Summarization. MIT Press, Cambridge (1999)
17. Moens, M., Boiy, E., Palau, R.M.: Automatic Detection of Arguments in Legal Texts. In: International Conference on artificial intelligence and law, Palo Alto, USA (2007)
18. Quinlan, J.R.: Induction of Decision Trees. Machine Learning 1(1), 81–106 (1986)
19. Ricardo, B., Berthier, R.: Modern Information Retrieval. Addison-Wesley, New-York (1999)
20. Riloff, E.: Automatically constructing a dictionary for information extraction tasks. In: Proceedings of the Eleventh Annual Conference on Artificial Intelligence, pp. 811–816. AAAI press/ MIT press (1993)
21. Sheridan, P., Smeaton, A.: The application of morphosyntactic language processing to effective phrase matching. Information Processing & Management 28(3), 349–369 (1992)

Early Bug Detection in Deployed Software Using Support Vector Machine

Saeed Parsa, Somaye Arabi Nare, and Mojtaba Vahidi-Asl

Department of Computer Engineering, Iran University of Science and Technology, Tehran, Iran
Parsa@iust.ac.ir, {Atarabi,Mojtaba_vahidi@comp.iust.ac.ir}

Abstract. Software crashes may be disastrous and cause great economical damages. Therefore, the reliability and safety of software products in some circumstances may be very vital and critical. In this paper a new mechanism to detect errors and prevent software crashes at run time, is presented. The novelty of the proposed technique is the use of Support Vector Machine (SVM) method to accelerate the detection of bugs early before they cause program crashes. By applying the SVM method, two thoroughly distinguishable patterns of failing and passing execution of the program are constructed in a relatively short amount of time, before the program is actually deployed. The vectors are constructed from the decision making expressions or in other words predicates, appearing within the program text. These patterns are further applied, after the program deployment, to estimate the probability of program failure symptoms, early before the program crashes. Our experiments with bug prediction in Siemens software, demonstrate the ability of our proposed technique to predict errors before they can cause any damages.

Keywords: Software Debugging, Early Bug Detection, Deployed Software, Support Vector Machine, Predicate.

1 Introduction

The more complicated the software, the more difficult it will be to debug. This reveals the need to develop automated software debugging tools [1]. One of the important fields in this context is debugging deployed software [2]. Bug localization is hard in these systems because there is no mechanism to inform the user when the software behaves anomalously. The only sign of anomalous behavior is a crash in software or undesirable output [3]. Existence of bugs in some vital deployed software may cause critical and deadly outcomes. For instance NASA Mars Global Surveyor battery failure was the result of a series of events linked to a computer error made five months before [19]; or because of a fault in software control of safety-critical systems such as The Therac-25 (Radiation therapy machine) accidents (1985-1987), at least five patients died [20].

Current automated debugging techniques develop a profile for a program's execution either through static inspection or dynamic instrumentation [4]. A static analysis detects program bugs after checking the source code using a well-specified program

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 518–525, 2008.

model (such as control flow graph) [5]. A dynamic analysis, usually tries to locate defects by contrasting the runtime behavior of correct and incorrect executions [6].

Dynamic techniques are based upon analysis of predicates. Predicates are simple Boolean expressions at various program points. Predicates are designed to capture potentially interesting program behaviors such as results of function calls, directions of branches, or values of variables [7, 8, 9]. To collect such information, extra code is inserted before each predicate within the program code. This process is called instrumentation [10]. During the program execution, the number of times each predicate is observed to be true or false is counted. This information is analyzed later to find potential bugs. These techniques entail the complete execution of the program, and hence could not be useful for early bug detection in deployed software.

One of the efficient approaches to debug deployed software is to employ models based upon learning algorithms. In these approaches a learning model which represents program behavior is constructed before deployment of the software. This model (pattern) is used later to detect anomalous behaviors of the program and informs the user about the existence of error-prone code [11, 12].

This paper presents a new machine-learning technique to detect anomaly dynamically while the program is executing. The technique employs a machine learning method called Support Vector Machine (SVM) [13], to build a model of a program behavior according to passing and failing executions of the program. It then uses the model to identify error-prone points. The distinguishing feature of our suggested approach is to detect anomaly dynamically during program execution and to find the location of bug before system crashes, in a relatively small amount of time.

The remaining part of this paper is organized as follows: Section 2 discusses previous approaches for debugging deployed software. In section 3, we introduce our proposed method for early bug detection. Section 4 evaluates functionality of proposed method in two case studies and includes experimental results. We conclude with final remarks and portray future work in section 5.

2 Related Work

Few works have been done on early bug detection in deployed software. Statistical debugging is one of the strongest dynamic methods within the software engineering field [4, 7, 8, 14]. It gathers information about program variables after the program is instrumented. The statistical data collected from program execution are formulated into a report containing specific execution data and parameters; this is referred to as a test case. Statistical debugging techniques take such set of test cases and apply an algorithm to determine which predicates are responsible for the programs failure. The applied algorithm generally uses a number of metrics dependent on correct and incorrect test cases [14]. Statistical debugging techniques depend upon the information which is completed after the program execution and therefore they cannot be applied dynamically while the program is executing. These techniques cannot detect anomalous or failing behavior as the software is executing. Our technique tries to find bugs before they really occur and informs user before software crashes.

In [15] a completely different approach for anomaly detection in deployed software is proposed. The tool which is introduced in this approach is called Diduce which is

inspired from Daikon [16]. The tool extracts properties of the program which are generated according to passed test cases and are called invariants. Invariants are constant properties of a program which are satisfied in all successful executions of the program. The generated invariants could be used to find the locations of potential defects if they are violated in failing test cases. This technique tries to debug deployed software. The drawback of this technique is its high overhead on executing program which is not desirable in deployed systems. Our proposed method produces very low overhead on executing software, because it uses simple vectors and SVM method which we employ for classification performs classification very fast and precisely.

In [11] a collection of statistical data is gathered based on predictive properties in program such as branches in order to understand program behavior and fault detection. This data is used to build behavior models by applying statistical machine learning techniques. The technique builds markov model for failing and passing executions of a program and then classify them based on the result of the execution. Program behaviors could be predicted based on this classifier. But the technique is not suitable for deployed software products because it needs complete program execution to build markov model.

3 Anomaly Prediction

Early bug detection can be performed in two main phases of training and deployment. As shown in Figure 1.a, the training phase consists of three steps. Figure 1.b shows the deployment phase. The details of these two phases are further discussed in this section.

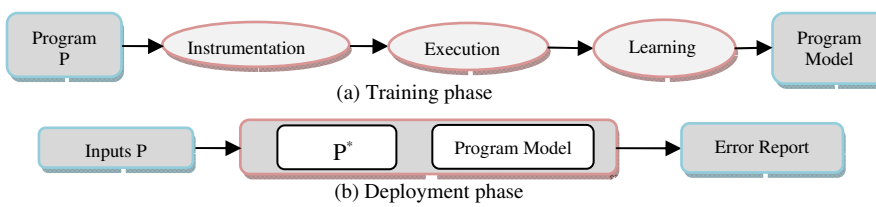


Fig. 1. A view of our proposed approach: (a) Training phase (b) Deployment phase

3.1 Training Phase

In the training phase a model based on passing and failing executions of a given program code is built. Training phase consists of three major steps: instrumentation, execution, and learning. In the instrumentation step of a program P, probes are inserted before the branch statements or the locations within the program code where the value of predicates may change, to generate the instrumented program P*. During the Execution step, P* is executed on the failing and passing test cases. For each instrumentation point a separate predicate vector is built. These predicate vectors are input to the learning step. In this step a model, representing the program behavior is built. The important steps of the training phase are execution and learning. These steps are further described below.

Execution. In the execution step of the training phase, the failing and passing test cases are applied to run the instrumented program code, several times. At each run for each instrumented point within the program a separate vector is built. Within a loop construct such as For and While, for each iteration of loop, a separate predicate vector including the loop iteration number, should be created for all the instrumented statements appearing within the loop body.

Each cell of the vector represents the value of a predicate at the time execution reaches the corresponding instrumented point. The vectors are classified in two groups of Fail and Pass which depends on whether the program fails or executes successfully. Based on this observation that errors most often are originated at predicates [10], we decided to select some of the critical predicates as instrumented points. Here, the criticality of a predicate is calculated as the number of the program branches rooted at the predicate. If the predicate at an instrumented point includes either $\leq$or $\geq$, we split it into two predicates. For example the predicate $A \leq B$ is converted into two predicates A<B and A=B. A sample code and its predicate vectors are shown in Figure 2. The program has three instrumented points (i.e. line numbers 3, 7, 11).

The right side of the table shows the predicate vectors produced at each instrumented point for four different inputs. For first two inputs the result of the method is pass and for last two outputs the result is fail. There are three predicates in the program: (a<=b), (c<d), (big1<big2); but as mentioned before we split predicates such as '<=' into two predicates '<' and '=', thus we would have six distinct predicates to build our vector: ((a<b), (a=b), (c<d), (c=d), (big1<big2), (big1=big2)) we call this vector, predicate vector. The elements of the vector are updated during the execution of the method. For input 1:(1,2,3,4), when execution reaches line 3 the vector value becomes (1,0,2,2,2,2) because in this line the predicate (a<b) is evaluated as true and predicate (a=b) is evaluated as false. All other elements have the value '2' because their corresponding predicates have not been executed. The updating process continues until at line 11 it becomes (1,0,1,0,1,0).

The class label of vector values for each instrumented point is the result of the method for that specific execution. To this end we use "1" for passing executions and "-1" for failing ones. At this point each instrumented point has one vector value for one execution of the method.

Learning. The learning step consists of building a model from predicates evaluation. We employ Support Vector Machine as a classifier to create the model [18]. First we use the provided vector values as the input to the classifier. SVM then performs classification according to the program executions results (i.e. fail and pass) obtained from previous step to make the model.

Modeling with Support Vector Machine, in our application in this paper, l training examples are obtained from *l* executions of the program. To this end, the program should be executed *l* times for different passing and failing test cases and for each instrumented point one individual classifier is constructed. All predicates in a method construct the structure of a vector. In other words each element of the vector corresponds to a predicate in that method. By execution of each instrumented point in the method, the value of each element of the vector is evaluated based on the value of predicates before that program point including the point itself. Thus for each instrumented point in the program, during *1* execution, we get *l* training example. In this application, class labels '-1' and '1' represent failing and passing execution of the program, respectively.

Line#	Sample code	Vector State ((a<b), (a==b), (c<d), (c==d),(big1<big2),(big1==big2))			
		Input1(1,2,3,4)	Input2(3,2,4,5)	Input3(4,3,2,4)	Input4(5,1,2,3)
1	Public class Max {				
2	Public static void max (int a,int b,int c,int d){				
3	If (a <= b)	(1,0,2,2,2,2)	(0,0,2,2,2,2)	(0,0,2,2,2,2)	(0,0,2,2,2,2)
4	Big1=b;				
5	else				
6	Big1=b;//incorrect				
7	If (c < d)	(1,0,1,0,2,2)	(0,0,1,0,2,2)	(0,0,1,0,2,2)	(0,0,1,0,2,2)
8	Big2=d;				
9	else				
10	Big2=c;				
11	If (big1<big2)	(1,0,1,0,1,0)	(0,0,1,0,1,0)	(0,0,1,0,0,1)	(0,0,1,0,1,0)
12	System.out.println("max is :"+big2)				
13	else				
14	System.out.println("max is :"+big1)				
15	} }				
	Program Result	pass	pass	fail	fail

Fig. 2. A sample program (left of table) with test inputs and predicate vectors produced by the program with those inputs (right of table)

We employ Support Vector Machine learning method with some training examples for each instrumented point of the program to build a model for that point. This model could be used after deployment to determine the probability that a vector instance (i.e. a set of values for the predicates) belongs to the class of '1' or '-1'.

3.2 Deployment Phase

As shown in figure 1, during the deployment phase, the program model is deployed with the instrumented program P* to simultaneously detect anomalies and identify their location in program. In our technique a statement is anomalous if it is classified as failing by SVM tool in execution time.

Early bug detection using SVM. After the model has been deployed with the instrumented software, the SVM model for each instrumented point of program under test, can classify that point to determine the effectiveness of that point on causing error. To classify the vectors of an instrumented point, SVM builds an optimal hyper plane which is written as equation (1):

$$f(x) = sign \sum_{i}^{l} (\alpha_i \, y_i \, \mathrm{K}(X_i, \mathrm{X}) + b) \qquad (1)$$

Where α_i is the Lagrange multiplier corresponding to each constraint, and $\mathrm{K}(X_i, \mathrm{X})$ is called a kernel function, it calculates similarity between two arguments X_i and X. SVM estimates the label of an unknown example whether sign of $f(x)$ is positive or not. In our application, example X is a vector for an instrumented point of program that the value of this vector is evaluated during a program run but the class of it is unknown and x_i is predicate vector for the same point in execution i during the training phase. According to the similarity between vector X and vectors x_i the class of can be determined.

4 Empirical Results

To evaluate the effectiveness of our approach, we implemented a prototype and conducted two case studies. The prototype consists of two parts: instrumentation and learning. We used the Aristotle analysis system to instrument the program [17]. The machine-learning part is implemented in C language and consists of two modules: learning and error detection. In learning module we build SVM models for each instrumented point. Building SVM has done by using LIBSVM [13]. LIBSVM is integrated software for support vector classification, regression and distribution estimation. It supports multi-class classification. The error detection module is used when the model is deployed within the program.

For the object of analysis, we used the Siemens suite [5]. The Siemens suite contains seven programs, faulty versions of those programs, and test suites designed to test those programs. Each faulty version contains exactly one fault, although the faults may span multiple statements or even functions. To construct powerful model which contains both passing and failing executions of the software, the fault seeding technique is employed.

The goal of case study 1 has been to determine the run time error detection capability of our approach. First each version is instrumented. Then, we run the instrumented version with passing and failing predefined test cases. The graph in Figure 3 shows the results of this study.

As shown in the graph increasing in number of test executions makes SVM model more powerful. It becomes trivial after looking at the graph that the percentage of errors which model can detect in unknown executions depends on the number of executions before deployment. The results show that the number of executions that is needed for precisely bug detection in our approach is smaller than the statistical approaches [7, 8].

The goal of case study 2 has been to determine the correctness of the error detection technique. To this end, we have studied two factors: False positive and false negative rates. The results in this study show that our approach is pessimistic and reports error to the user in some cases that there is no real error. But the percentage of errors that are not detected by our approach is low.

Fig. 3. Percentage of error detected based on passing and failing execution for building model

5 Conclusion and Future Work

In this paper, we have presented a novel machine learning technique for early bug detection in deployed software. We have employed Support Vector Machine to build a model of a program before deployment. We use this model to classify each running statement after deployment as a cause of program failing or passing. The first novelty of our work is its fastness and lower overhead on deployed software. Another novelty is the way that the technique informs the user about the anomalies before they cause failure in the program. The empirical results show that our proposed technique is effective in detecting most of suspicious statements of the program.

For future work other learning methods could be applied on early bug detection in deployed software. We can also use ensemble methods to improve the accuracy of the results. The efficiency of the method could be improved if we decrease the overhead of instrumentation using lightweight instrumentation. Another efficiency we concern about is trying to improve the accuracy of error reports in order to decrease time and effort of user for debugging.

References

1. Renieris, M., Reiss, S.P.: Fault localization with nearest neighbor queries. In: 18th IEEE International Conference on Automated Software Engineering, Montreal, pp. 30–39 (2003)
2. Carzaniga, A., Fuggetta, A., Hall, R.S., Van Der Hoek, A., Heimbigner, D., Wolf, A.L.: A Characterization Framework for Software Deployment Technologies. Technical Report, CU-CS-857-98, Dept. of Computer Science, University of Colorado (1998)
3. Bowring, J., Orso, A., Harrold, M.J.: Monitoring deployed software using software tomography. In: ACM SIGPLAN-SIGSOFT workshop on Program analysis for software tools and engineering (PASTE 2002). Software Engineering Notes, vol. 28(1), pp. 2–9. ACM Press, Charleston (2002)
4. Liu, C., Yan, X., Fei, L., Han, J., Midkiff, S.P.: Sober: Statistical model-based bug localization. In: 10th European Software Engineering Conference/13th ACM SIGSOFT Int'l Symposium Foundations of Software Eng (ESEC/FSE 2005), Lisbon, pp. 286–295 (2005)
5. Hutchins, M., Foster, H., Goradia, T., Ostrand, T.: Experiments of the effectiveness of dataflow- and control flow-based test adequacy criteria. In: 16th International Conference on Software Engineering, Sorrento, pp. 191–200 (1994)
6. Cleve, H., Zeller, A.: Locating Causes of Program Failures. In: 27th Int'l Conference Software Engineering ICSE 2005, St. Louis, pp. 342–351 (2005).
7. Liblit, B., Naik, M., Zheng, A., Aiken, A., Jordan, M.: Scalable Statistical Bug Isolation. In: ACM SIGPLAN 2005 Int'l Conference Programming Language Design and Implementation (PLDI 2005), Chicago, pp. 15–26 (2005)
8. Zheng, A., Jordan, M., Liblit, B., Naik, M., Aiken, A.: Statistical debugging: simultaneous identification of multiple bugs. In: 23rd international conference on Machine learning (ICML 2006), pp. 1105–1112. ACM Press, New York (2006)
9. Jones, J., Harrold, M.J.: Empirical evaluation of the tarantula automatic fault-localization technique. In: 20th IEEE/ACM International Conference on Automated Software Engineering, Long Beach (2005)

10. Santelices, R., Sinha, S., Harrold, M.J.: Subsumption of Program Entities for Efficient Coverage and Monitoring. In: Third International Workshop on Software Quality Assurance (SOQUA 2006), Portland, pp. 2–5 (2006)
11. Bowring, J., Rehg, J.M., Harrold, M.J.: Active Learning for automatic classification of software behavior. In: International symposium on software testing and analysis, Boston, pp. 195–205 (2004)
12. Brun, Y., Ernst, M.: Finding Latent Code Errors via Machine Learning over Program Executions. In: 26th Int'l Conference on software Engineering (ICSE 2004), Edinburgh, pp. 480–490 (2004)
13. Chang, C.-C., Lin, C.-J.: LIBSVM - A Library for Support Vector Machines, http://www.csie.ntu.edu.tw/~cjlin/libsvmtools
14. Liblit, B.: Cooperative Bug Isolation. PhD thesis, University of California, Berkeley (2004)
15. Hangal, S., Lam, M.: Tracking down software bugs using automatic anomaly detection. In: 24th international conference on software engineering, Orlando, pp. 291–301 (2002)
16. Ernst, M., Cockrell, J., Griswold, W.G., Notkin, D.: Dynamically discovering likely program invariants to support program evolution. In: 21st International Conference on Software Engineering, pp. 213–225. ACM Press, Los Angeles (1999)
17. Harrold, M.J., Larsen, L., Lloyd, J., Nedved, D., Page, M., Rothermel, G., Singh, M., Smith, M.: Aristotle: A system for development of program analysis based tools. In: 33rd Annual ACM South east Conference, Clemson, pp. 110–119 (1995)
18. Gunn, S.R.: Support Vector Machines for Classification and Regression. Technical Report, Faculty of Engineering, Science and Mathematics School of Electronics and Computer Science, university of southampton (1998)
19. National Aeronautics and Space Administration, http://www.nasa.gov/mission_pages/mgs/mgs20070413.html
20. Leveson, N., Turner, S. Clark: The Therac-25 Accidents, http://courses.cs.vt.edu/~cs3604/lib/Therac_25

A New Architecture for Heterogeneous Context Based Routing

Enrico Dressler, Raphael Zender, Ulrike Lucke, and Djamshid Tavangarian

University of Rostock, Department of Computer Science,
Chair of Computer Architecture, Rostock, Germany
firstname.lastname@uni-rostock.de

Abstract. This paper presents a new architecture for a heterogeneous context based routing system called HCBR that classifies devices with homogeneous network interfaces into horizontal cells and uses devices with interfaces in different cells to combine these cells to a heterogeneous ensemble by using vertical communication structures. Furthermore, a central and a decentral approach of this HCBR architecture are described which allow a flexible device communication inside an ensemble and permit reliable and context based data exchange between devices of different cells. Additionally, requirements in Quality of Service (QoS) as well as context of applications are considered for selection of suitable horizontal and vertical communication channels in a highly heterogeneous network infrastructure of a pervasive environment.

Keywords: Heterogeneous network architecture, gateway, routing, WLAN, Bluetooth, QoS.

1 Introduction

Within the next years the continuing technological progress in microelectronics with more and more decreasing device dimensions will make available small, spontaneous, and wireless communicating devices. These devices support users in their everyday work. Since computers of this type are getting smaller, cheaper, and increasingly powerful they can be hidden in objects of our everyday life. Generally, they have the technical potential to create a communication network and communicate together. They join when needed, not necessarily supported by an existing infrastructure or any other kind of fixed base stations. Because nodes are moving freely, the network topology changes dynamically in an unpredictable manner. In contrast to classical network architectures such networks gain significance and enable to realise an environment for ubiquitous information processing which focuses the proactive support of the users in their activities.

Wireless communications, including Wireless Personal Area Networks (WPAN), Wireless Local Area Networks (WLAN), Wireless Metropolitan Area Networks (WMAN), Wireless Regional Area Networks (WRAN) as well as cellular wide area networks and satellite networks, enjoy an unprecedented growth over recent years so that millions of people can exchange information every day using mobile phones, notebooks, personal digital assistants (PDAs), sensors in their environment and other

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 526–534, 2008.

wireless based products. This widespread and integrated use of wireless networks enable devices to use a specialised wireless communication for a certain task considering Quality of Service (QoS) requirements like bandwidth, delay, jitter, costs, and energy consumption.

In a highly mobile pervasive environment WPANs that have been designed to exchange information between personal objects within the short-range of an individual play a decisive role. They can be used to replace wires between computers and their peripherals, to share multimedia content amongst devices, and to build an infrastructure for sensor networking applications. Bluetooth and ZigBee are the best examples representing WPANs. A key feasibility issue of WPANs is the inter-working of wireless technologies to create a heterogeneous wireless environment. For instance, devices interconnected in a WPAN may be able to utilize a combination of 3G access and WLAN access by selecting the one of which the requirements are most suitable. Moreover, WPANs and WLANs will enable an extension of the cellular network into devices without direct cellular access. In such networks, different wireless technologies do not compete, but complement each other to use the best connectivity for the current purpose. The task of adaptable heterogeneous networks that provide QoS guarantees to users is extremely complex and challenging [1].

This paper presents a concept for an architecture that provides an integration of heterogeneous access technologies, such as cellular, WLAN, WMAN, WPAN, and sensor networks whereas the complexity of such a heterogeneous environment is hidden not only from end users, but also to be made transparent to applications.

Section 2 classifies different access technologies into homogeneous cells and heterogeneous ensembles. Then section 3 illustrates two different approaches for a system that combines these cells to an ensemble. The centralised approach uses a common characteristic of a single hop operation mode, where devices can access the system through a fixed General Purpose Access Point (GPAP) that supports different communication technologies simultaneously and is connected to a wired or wireless infrastructure backbone. The decentralised approach distributes the bridging functionality of the fixed General Purpose Access Point to devices like PDAs or notebooks that already exist in a pervasive environment. Thereby, the heterogeneous communication can be extended to a multi-hop communication environment that provides alternative connections inside an ensemble, can use central infrastructure, and also allows communication without fixed base stations. This distributed GPAP is an efficient approach to build a pervasive environment that provides seamless integration of one-hop networks (e.g., cellular, WRAN) and multi-hop wireless systems (e.g., WLAN, Bluetooth, ZigBee), and supports the use of communication channels depending on application needs and available types of radio access networks. Afterwards section 4 proposes an architecture that combines the homogeneous cells to a heterogeneous ensemble and provides an efficient support for the integration of WPANs, WLANs, WMANs and WRANs. The support of context-aware applications running on ensemble members in a heterogeneous environment will be explained at the end of this section. In collaboration with a context-aware service middleware, designed at the University of Rostock [3], it will be possible to obtain application context to the proposed distributed GPAP architecture and to consider it for selection of suitable communication channels to fulfil the application needs in a heterogeneous network. When all these technologies are integrated with the Internet, the possibilities

are countless, but unfortunately, the processes to achieve this are nontrivial [2]. At the end of this paper related work in the area of inter-working of WLAN and WPAN technologies as well as an approach for future pervasive environments are presented. The paper concludes with a summary.

2 From Cells to Ensembles

A pervasive environment originates a typical heterogeneous network with horizontal and vertical network structures, displayed in Fig. 1.

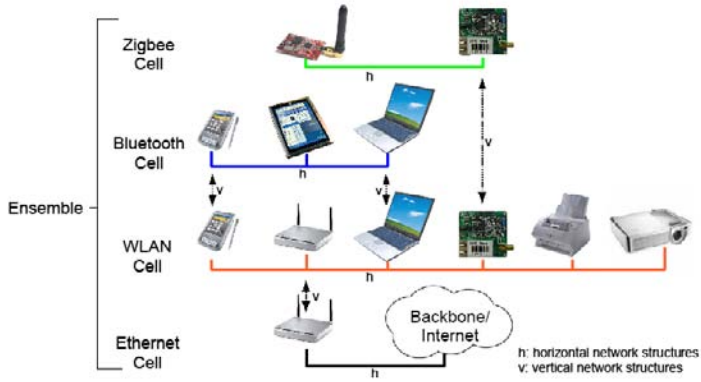


Fig. 1. Horizontal and vertical network structures

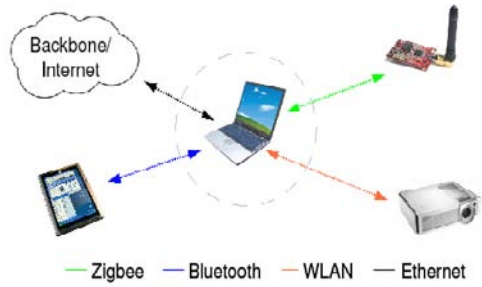


Fig. 2. Centralised approach for a GPAP

We define a horizontal network structure as a locally closed collection of devices interconnected by the same communication technology. Devices in these cells can be interconnected wired (e.g., Ethernet, USB, Serial) or wireless (e.g., WLAN, Bluetooth, ZigBee). For example, a PDA, a Tablet-PC and other Bluetooth-capable devices build a common Scatternet. In another cell WLAN-capable devices build an ad-hoc network. Wireless cells are dynamic because devices enter or leave the cells

scope spontaneously. Wired cells may be dynamic as well, because they can be powered on/off or cables can be plugged or unplugged unpredictable.

Devices like the PDA or the notebook with network interfaces of different cells can be used to combine these horizontal structures by adding vertical structures. Thus, devices that utilise more than one communication technology bridge between cells, depending on their communication technologies, to combine the cells to a heterogeneous ensemble. The ensembles structure can alter spontaneously dependent on the underlying dynamic cells.

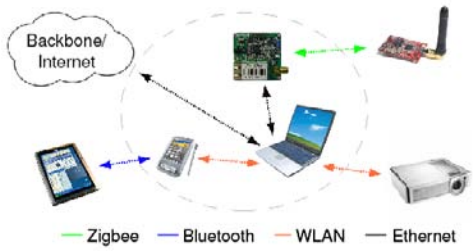


Fig. 3. Decentralised approach for a GPAP

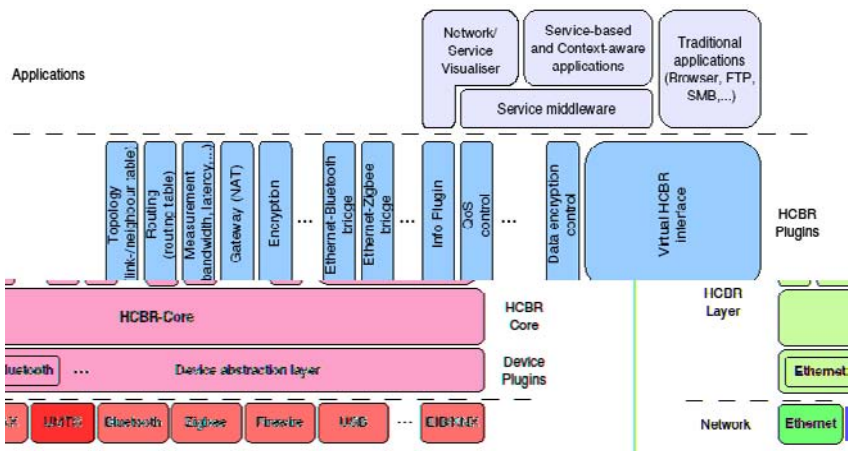


Fig. 4. Structure of the HCBR architecture

3 Centralised vs. Decentralised Approach

For a system that combines homogeneous cells to a heterogeneous ensemble, two different approaches are possible. The first approach, displayed in Fig. 2, is a centralised approach. This approach is made up by a device (notebook) with numerous network interfaces (Ethernet, WLAN, ZigBee, Bluetooth), each representing a member of another horizontal network structure. Thus, the notebook can send video data to the projector to make a presentation by using WLAN, share data with the TabletPC by using Bluetooth, obtain sensor data by using ZigBee, connect the different

homogeneous cells with each other to allow communication between them, and connect all the devices to the Internet using a wired connection.

The second approach, displayed in Fig. 3, is a decentralised approach. This approach uses already existing devices of a pervasive environment. Each of them with just a couple of different network interfaces.

In this approach the functionality of combining horizontal cells to a heterogeneous ensemble is distributed to several devices. Thus, a PDA with Bluetooth and WLAN interfaces can build a vertical network structure and bridge between Bluetooth and WLAN cells. Another bridge can be established by a device with ZigBee and Ethernet, or a notebook with WLAN and Ethernet interfaces. Here, the collaboration of several devices with a few different interfaces can achieve the functionality of the centralised approach as well, but without the need of specialised devices with a vast number of network interfaces. The decentralisation avoids a single point of failure, represented by a central device, as well.

4 HCBR Architecture

In a pervasive environment a most flexible device communication is required that permits reliable and context-aware data exchange with integration of all kinds of wired and wireless technologies. Thereby, quality requirements in data exchange between mobile devices and sensors in the environment, as well as context of applications are to be considered for selection of suited communication channels in a highly heterogeneous network infrastructure (Bluetooth, WLAN, WiMAX, UMTS, GPRS, Ethernet, EIB/KNX, …). Fig. 4 illustrates the concept of an architecture for heterogeneous context based routing (HCBR) which enables communication of this type by using horizontal and vertical structures to organise homogeneous cells to an heterogeneous ensemble.

The main idea of the HCBR architecture is made up by a device with multiple network interfaces. These interfaces can use different technologies for communication so that each interface can be a member of a different horizontal cell. Thus, the device is a kind of general purpose access point that can communicate with all members in each cell. Until now, devices can just communicate with devices in their homogeneous cell. To communicate with devices in other cells, it is necessary to add vertical network structures and bridge the communication between the different cells. By combining the horizontal network structures in such an overlay manner, multiple homogeneous cells are combined to a heterogeneous ensemble, in which devices of different cells can communicate with each other. Hence, the HCBR architecture exceeds the possibilities of a general purpose access point.

The HCBR core provides the central point for this architecture. It supports a modular enhancement by a plugin concept to support new communication technologies and functionalities. The HCBR core can be extended by two types of plugins: Device-Plugins and HCBR-Plugins, and is responsible for their collaboration.

Device-Plugins serve as enhancement for the architecture in order to support new network interfaces. For every new technology a Device-Plugin is necessary that encapsulates special characteristics for arranging the communication with the specific network interface. Characteristics in data transport are treated as well. Thus, Device-Plugins provide an intermediate layer to abstract from available wired (e.g., Ethernet,

USB, Serial, EIB/KNX) and wireless (e.g., WLAN, WiMAX, UMTS, Bluetooth, ZigBee) network interfaces in the operating system.

HCBR-Plugins are used to enhance the HCBR core with new functionalities. The most important functionality is the realisation of vertical network structures to bridge between different homogeneous cells. To reduce the amount of bridges needed, we use Ethernet as the general technology and implement bridges between Ethernet and other technologies only. Thereby, we need to create bridges for e.g., Bluetooth-Ethernet, ZigBee-Ethernet, EIB/KNX-Ethernet, but do not need bridges like Bluetooth-ZigBee, because this functionality can be composed by a Bluetooth-Ethernet and a ZigBee-Ethernet bridge as well.

Furthermore, HCBR-Plugins are needed to distribute the GPAP functionality on different devices with a few couple of network interfaces, illustrated in Fig. 3. Therefore a topology plugin is necessary that acts like a proactive routing protocol [7] to gather and exchange information on neighbours, links, and network topology and provides it to other HCBR-Plugins. Another plugin, which is responsible for routing, can use these information to build a routing table by using a metric for heterogeneous networks that uses the available network capacity of many channels in a smart way to significantly improve the overall capacity of the network. Hence, measured-based [10] and status-based methods [11] can be used to realise multi path routing. Another plugin uses the information of the topology plugin to suitably coordinate frequency bands in a wireless local area network with autonomous and dynamic network adaptation depending on an application's context. Therefore it can use flexible approaches that allow each device to potentially access all the channels by switching some of its interfaces among the available channels dynamically [12,13] based on node density, traffic, and channel conditions that has shown to be a good choice in theory [14,15].

In order to allow traditional applications to abstract from the complexity of multiple different interfaces in a heterogeneous network and to use some functionality of the HCBR architecture, we use a single virtual network interface (also used in [16]), implemented by a so-called "Virtual-HCBR-interface". Thus, standard applications can use features of the HCBR architecture transparently.

Plugins for measurement of bandwidth and latency are required to use routing metrics like Weighted Cumulative Expected Transmission Time (WCETT) [17] for calculations in the Routing-Plugin, create gateway functionality, support MobileIP [18], and allow data encryption.

Furthermore, an Info-Plugin is required that passes information from the HCBR core to a graphical user interface in order to allow analysis of real network behaviour.

A HCBR-Plugin for context-awareness is planned to allow an optimal collaboration between a service based middleware [3], designed at the University of Rostock, and context based routing with HCBR. We use context in terms of QoS requirements between ensemble members to select suitable communication channels in a highly heterogeneous network (e.g., a channel with low latency and jitter for Voice over IP data). In combination with the virtual network interface it will be possible for context-aware applications to use this interface as a standard, to obtain application context from the service based middleware und use the QoS-Plugin to commit QoS requirements like bandwidth, latency, or data encryption to the HCBR core. Finally the HCBR core is able to consider context information in form of QoS requirements for selection of suitable communication channels to fulfil the applications needs in a heterogeneous network.

5 Related Work

A key feature of the proposed architecture is the inter-working of WPAN and WLAN technologies to create a heterogeneous environment. Given the importance within the WPAN operating space, availability of devices, and intensive research activities, the industry standard Bluetooth is the best example for a WPAN. The Bluetooth WPAN technology defined by the Bluetooth Special Interest Group (SIG) [4] is a robust and flexible low cost and short-range radio communication standard that can be found in many consumer electronics such as cell phones, PDAs, and notebooks.

To carry IP packets over Bluetooth, two possible protocol stacks can be used. The first option is to transmit IP packets over the point-to-point protocol (PPP) over Radio Frequency Communication (RFCOMM) [8], by taking advantage of the fact that PPP is already implemented in most mobile devices. Due to the fact that such an arrangement has been found to be highly inefficient [8] the Bluetooth SIG has published a native way for carrying IP traffic over Bluetooth by a protocol called Bluetooth network encapsulation protocol (BNEP) wherein IP packets are encapsulated in Ethernet packets which are carried over Bluetooth links. This ability to communicate with a LAN allows WPANs to take advantage to services such as printing, Internet access and file sharing. Therefore, a device that is Bluetooth and LAN protocol aware can be used as gateway to the LAN. The connection procedure and setup time is extremely low and is therefore a very attractive option, considering the high data rates of 11-55 Mbps in traditional WLANs. The IEEE 802.15 Working Group is investigating in protocols to access an 802.11 WLAN directly, but the need for additional hardware may impact the size and power constraints of WPAN devices.

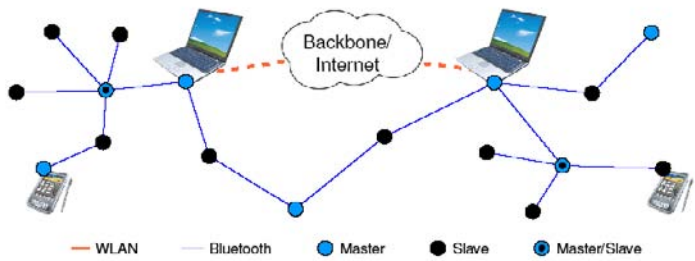


Fig. 5. The use of WLAN as a Bluetooth backbone

Current developments like BlueStar [5] or [6] enable efficient integration between Bluetooth WPANs and IEEE 802.11 WLANs. They use a few Bluetooth Wireless Gateways (BWGs), that are also IEEE 802.11 enabled so that they can serve as access point to the IEEE 802.11 wireless network. BlueStar enables low-cost and short-range Bluetooth devices, belonging to either a piconet or a scatternet to access the global Internet infrastructure by utilising WLAN-based high-powered transmitters. The BlueStar architecture of Bluetooth and WLAN enabled devices is an intuitive and practical solution to this ad hoc issue.

6 Conclusions and Future Work

An architecture like BlueStar is necessary to allow Bluetooth devices to access the Internet but the need for an IP address and IP stack on every Bluetooth device is a big drawback of this approach. For Bluetooth devices that do not bring their own IP address and want to connect to a BWG a mechanism is needed to assign IP addresses and to provide the device with the relevant information for connecting to the network (DNS server, default router, netmask, etc.). This way is very complex, but it allows Bluetooth devices to exchange information in a BlueStar-based network. One major feature that Bluetooth is originally not depending on an IP address disappears.

In a heterogeneous environment a couple of devices use Internet access (e.g., PDAs, notebooks) but there are a high amount of other devices that only want to exchange data in their homogeneous cell including sensors in the environment of the user. This class of devices are characterised by a limited energy budget and should run as long as possible. Using an IP stack on such devices, that increases the data packet size, represents additional communication costs. For this reason an architecture for a heterogeneous network has to distinguish between devices that need and those that do not need Internet access and it has to support both device types. Using the approach of BlueStar for the proposed HCBR architecture seems to be efficient to fulfil the requirements of the first task. Leaving devices in their homogeneous cell and supporting assembly of multiple cells with the same technology by using an IP backbone, illustrated in Fig. 5 and realised by the proposed HCBR architecture, can solve the requirements of the second task.

PDAs, notebooks, sensors, and other devices in the environment of the user can use Bluetooth to build a Scatternet that enables direct data exchange inside the homogeneous Bluetooth cell. Furthermore, the notebooks have WLAN interfaces that offer higher bandwidth and lower latency in a WLAN cell. This cell can be used to build a fast backbone for the Bluetooth network that accelerates the communication between the PDAs. In this approach the Bluetooth packets must be sent from the PDA to the notebook, tunnelled within WLAN packets, transmitted to the other notebook, unpacked, and sent to the other PDA using Bluetooth. Such a communication needs to be made transparent for the PDAs to allow a real heterogeneous network. This approach offers new possibilities for Bluetooth and ZigBee enabled devices.

7 Summary

This paper presents a new architecture for a heterogeneous context based routing system called HCBR that uses horizontal and vertical network structures to combine homogeneous cells to heterogeneous ensembles. Furthermore, it presents a centralised and a decentralised approach for this HCBR architecture which allows a flexible device communication inside an ensemble and permits reliable and context based data exchange between devices of different cells. Additionally, requirements in QoS as well as context of applications are considered for selection of suitable horizontal and vertical communication channels in a highly heterogeneous network infrastructure of a pervasive environment. At the end of this paper related work in the area of interworking of WLAN and WPAN technologies as well as an approach for future real

pervasive environments with transparent use of different communication technologies are presented which are also part of our future research.

Acknowledgements

This research is supported by the German National Science Foundation (DFG, GRK1424). We would like to thank the anonymous referees for their valuable comments and suggestions.

References

1. de Morais Cordeiro, C., Agrawal, D.P.: AD HOC & SENSOR NETWORKS - Theory and Applications. World Scientific Publishing Co. Pte. Ltd., Singapore (2006)
2. Cavalcanti, D., Cordeiro, C., Agrawal, D., Xie, B., Kumar, A.: Issues in Integrating cellular Networks, WLAN, and MANETs: A Futuristic Heterogeneous wireless Networks. IEEE Wireless Communications 12(3), 30–41 (2005)
3. Zender, R., Dressler, E., Lucke, U., Tavangarian, D.: Meta-Service Organization for Pervasive Universities (unpublished)
4. Bluetooth Special Interest Group (SIG), Bluetooth specification, http://www.bluetooth.com
5. Cordeiro, C., Abhyankar, S., Toshiwal, R., Agrawal, D.: BlueStar: Enabling Efficient Integration between Bluetooth WPANs and IEEE 802.11 WLANs. ACM/Kluwer Mobile Networks and Applications (MONET) Journal (2004)
6. Lim, Y., Kim, J., Min, S.L., Ma, J.S.: Performance Evaluation of the Bluetooth-based Public Internet Access Point. In: Proceedings of the 15th International Conference on Information Networking (ICOIN) (2001)
7. Clausen, T., Jacquet, P.: Optimized Link State Routing, RFC 3626, IETF (2003), http://tools.ietf.org/html/rfc3626
8. Park, W., et al.: Specification of the Bluetooth System (July 2007), http://bluetooth.com/NR/rdonlyres/F8E8276A-3898-4EC6-B7DA- E5535258B056/6545/Core_V21__EDR.zip
9. Stuedi, P., Alonso, G.: Transparent Heterogeneous Mobile Ad Hoc Networks. In: ACM MobiQuitous (2005)
10. Liu, K., Li, J., Huang, P., Fukuda, A.: Adaptive Acquisition Multiple Access Protocol in Wireless Multihop Mobile Ad Hoc Networks. In: Proceedings of the 55th Vehicular Technology Conference (VTC) (2002)
11. Li, J., Haas, Z., Sheng, M., Chen, Y.: Performance Evaluation of Modified IEEE 802.11 MAC for Multi-Channel Multi-Hop Ad Hoc Networks (2003)
12. Raniwala, A., Gopalan, K., Chiueh, T.: Centralized Channel Assignment and Routing Algorithms for Multi-Channel Wireless Mesh Networks. Mobile Computing and Communications Review 8(2), 50–65 (2004)
13. Raniwala, A., Chiueh, T.: Architecture and Algorithms for an IEEE 802.11-Based Multi-Channel Wireless Mesh Network. In: IEEE Infocom (2005)
14. Kyasanur, P., Vaidya, N.H.: Routing and Interface Assignment in Multi-Channel Multi-Interface Wireless Networks. In: IEEE WCNC (2005)
15. Bahl, P., Chandra, R., Dunagan, J.: SSCH: Slotted Seeded Channel Hopping for Capacity Improvement in IEEE 802.11 Ad-Hoc Wireless Networks. In: ACM Mobicom (2004)

Performance Modeling of a Distributed Web Crawler Using Stochastic Activity Networks

Mitra Nasri, Saeed Shariati, and Mohammad Abdollahi Azgomi

Department of Computer Engineering,
Iran University of Science and Technology, Tehran, Iran
mitra_nasri@comp.iust.ac.ir, saeed_shariati@comp.iust.ac.ir,
azgomi@iust.ac.ir

Abstract. One of the basic requirements of Web mining is a crawler system, which collects the information from the Web. To predict the performance, dependability and other operational measures of a system, it is required to construct and evaluate a formal model of the system. We have constructed a formal model for a distributed crawler, which is based on UbiCrawler, using stochastic activity networks (SANs). The constructed SAN model is used to evaluate some performance measures of the crawler. The results of the evaluation of throughput are same as the published statistics of UbiCrawler. In addition, we have been able to evaluate two other measures that are communication overhead and coverage. In this paper, we will discuss the architecture of the distributed crawler. Then, we will present a SAN model of the crawler and the results of its evaluation.

Keywords: Web crawler, performance modeling, stochastic activity networks.

1 Introduction

Nowadays Web mining systems are critical to gather and analyze the structure of available information on the Web. A crawler is a software component that iteratively collects information from the Web, downloads pages and follows the linked URLs [7]. Output documents of a crawler, collected from the Web, can be used in many purposes such as Web mining [8]. On the other hand, the knowledge about Web structure will help us to design a more effective mechanism for crawling and indexing. For example, the ranking of pages may be useful to find and index high-quality URLs.

The continuous evolution of the Web and the forthcoming of new usage contexts confirm enduring research in crawling technology. This is interesting to know that founders of an old crawling system [15] are updating their crawler's body every 12 to 18 months to reflect changes in the structure of the Web.

Although an architecture of a crawler is theoretically easy to understand, its development is very expensive and time-consuming, involving the team to overcome many challenges when facing the changes of Web data or even new data types. As developers of Viuva Negra crawler [12] say, when a crawling machine leaves the experimental environment, many problems will arise. So there is a need to study all aspects of a crawler before starting the development phase.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 535–542, 2008.

Development of crawlers started from middle 1993 [14] but until 1998 when Google [7] introduced its distributed crawling module, all of the efforts were based on single crawlers. What exactly Brin's team suggested at 1998, was a centralized architecture to manage distributed crawling processes. This architecture may result in bottleneck on center node which should globally decide about arriving events although there were some distinct centralized processes for each task. After a while AltaVista search engine introduced its crawling module called Mercator [13] which was scalable (to search the entire Web) and extensible. But there was a basic change in crawler's main idea which made them distributed. A distributed crawler is a Web crawler that operates simultaneous crawling agents [9]. Based on this definition, UbiCrawler [6] were introduced as a fully distributed crawler containing some crawling agents each of them runs on a different computer. On every agent, limited numbers of threads running in parallel keep several TCP connections open at the same time. In principle some agents could be on different geographical locations [11] to improve downloading speed using agents close to the targets.

Finally, Viuva Negra [12] crawler as a part of Portuguese Web search engine called Tomba was introduced claiming to be fault-tolerant and scalable distributed crawler. There are many academic publications about the results which it collected during its development, including many statistical data of the Web. We use these results in next sections as input data for our model.

Due to expensiveness and time-consuming task of implementation of a distributed Web crawler [9], there is an essential need to model this system before doing anything. Any architectural decision on a crawler may affects on its performance and cost. Scalability and dependability of such system are also other reasons of needing to formal modeling. If a distributed crawler cannot manage most of faulty or hazardous states, it may not be useful in long time tasks of harvesting the Web. Unfortunately there was not any related work for modeling of a crawling system so the team decided to generate a model, based on one of recently introduced architectures, UbiCrawler and then, evaluate the results. Tuning stochastic parameters of the model is done using real statistical data [5, 12, 3] of Viuva Negra Crawler. We will discuss the details on the next sections.

In this paper we present a stochastic activity network [17] model of distributed crawling system and the results of its performance evaluation using Möbius tool [10]. The remainder of this paper is organized as follow. The architecture of Web crawler is introduced in Sec. 2. Some details of the model are discussed in Sec. 3. Main results of the evaluation of the model are presented in Sec. 4. Finally, some concluding remarks are mentioned in Sec. 5.

2 Architecture of Web Crawler

As stated before, the crawling module downloads and gathers relevant objects from the Web. Fig. 1 shows a simple algorithm of a Web crawler. A crawler should be distributed, efficient with respect to the use of the network, and prioritize downloading high quality objects [4].

	Require: *p*1, *p*2, ..., *pn* starting URLs
1	Q = {*p*1, *p*2, ..., *pn*}, queue of URLs to visit.
2	$V = \Phi$, visited URLs.
3	**while** $Q \neq \Phi$ **do**
4	Dequeue $p \in Q$, select *p* according to some criteria.
5	Do an asynchronous network fetch for *p*.
6	$V = V \cup \{p\}$
7	Parse *p* to extract text and outgoing links
8	$\tau^+(p) \leftarrow$ pages pointed by *p*
9	**for** each $p' \in \tau^+(p)$ **do**
10	**if** $p' \notin V \wedge p' \notin Q$ **then**
11	$Q = Q \cup \{p'\}$
12	**end if**
13	**end for**
14	**end while**

Fig. 1. Typical crawling algorithm [8]

The remainder of this section will discuss about some important parts of a Web crawler.

2.1 Partitioning Policy

A *distributed crawling system* needs a strategy to assign the URLs that are exposed by each agent to others, because the agent that discovers a URL may not be the one responsible for downloading it. All of the crawling agents must agree with such policy at the start of the task. To avoid downloading more than one page from each server simultaneously, the same agent is responsible for all the content of a set of Web servers in most distributed crawling systems [4]. Using locality of links (links on the Web pointing to other pages in the same server); an agent can save many unnecessary communications in sending URLs to other agents [6].

Partitioning policy should also be dynamic and can be adapted when new agents join or leave the system. A sample of such policy is *consistent hashing*, stated at Karger's work [16] which we use it as the base strategy in our model. This method is independent from the number of documents on servers and also can balance the load.

2.2 Dependability Challenges

Regardless of implementing a distributed or centralized Web crawler, there is an essential need to overcome connection failures during locating the site, opening an HTTP connection, downloading target file, converting it to HTML object and then parsing phase of one crawling task. None of these faults should stop the whole process. Therefore, a crawler must be fault-tolerant and reliable to do its mission in a reasonable time period. Dealing with distributed crawlers, there should be policies to re-organize partitioning method when an agent leaves the system. Fortunately consistent hashing can do this job well.

3 A SAN Model for a Distributed Crawler

The architecture of our proposed crawler is modeled by stochastic activity networks using Möbius tool [10]. Fig. 2 shows a global view of the model. The system is constructed from distinct agents each of them has finite number of threads. These threads, select a URL from their queue, download it, convert it to HTML, parse it to find new URLs, and then insert URLs of their own site into their own queue and leave the remained URLs for the agent. Using consistent hashing, an agent decides which URL should be sent to other agents. To know more about agent initiation and communications, please see Fig. 3 and Table 1.

Each thread downloads a URL according to the suggested as modeled in method in Fig. 4. As this figure shows, faults are the nature of this process and occur with different rates for different types of documents. The next process is conversion of downloaded MIME files to HTML, which has been shown in Fig. 5. Section 3.1 will discuss the rates and probabilities for these two processes. To complete remained statistics discussions on number of out-links of a page (used in analyzing process), please see section 3.2.

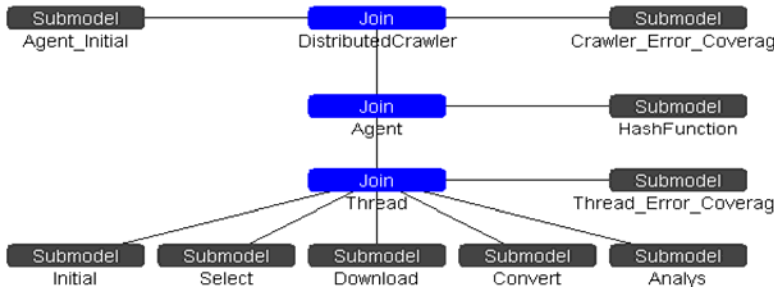


Fig. 2. Model global view (replicate/join hierarchy)

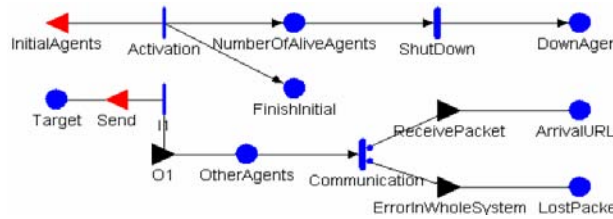


Fig. 3. Agent-Initial SAN sub-model including agent communications

3.1 File Downloading

More than 95% of the existing files on Web sites are HTML files [3]. Error rates in downloading and converting MIME files are different for each type so they should be studied separately. There is also a common source of downloading errors named

"Server Error", which is originated from Web server's behavior. For example, "Forbidden Server" is a server error which is independent from document types or DNS errors. Fig. 6 (a-c) shows MIME object's distribution and conversion error rate. Part (b) of this figure demonstrates server errors as stated above.

Table 1. Details of input gate guard functions for Agent-Initial sub-model

Name	Prediction	Function
Send	Target->Mark()>=PACKET_SIZE* (NumberOfAliveAgents->Mark()-1)	Target->Mark() -= PACKET_SIZE * (NumberOfAliveAgents->Mark()-1);
Initial Agents	FinishInitial->Mark() < NUM-BER_OF_AGENTS	
Receive Packet		ArrivalURLs->Mark() += PACKET_SIZE;

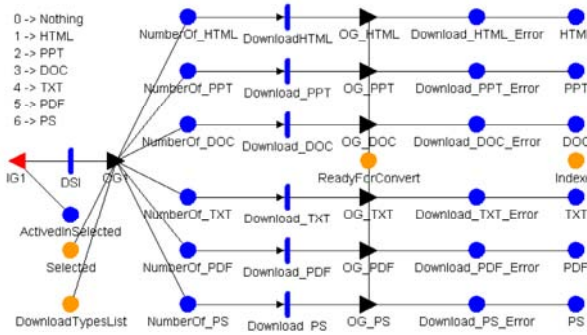


Fig. 4. Download SAN sub-model

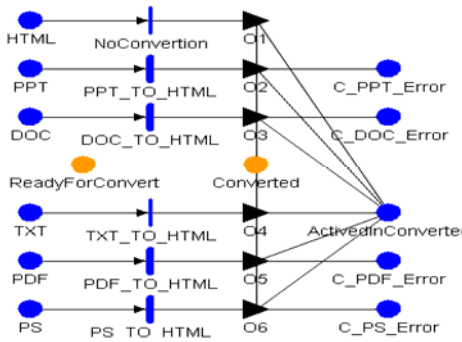


Fig. 5. Convert SAN sub-model modifies objects and generates corresponding HTML files

3.2 Statistics of URLs

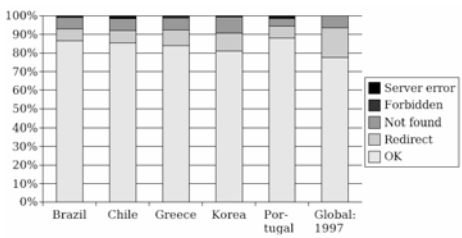
To start discussing about URLs statistics, we should primary introduce power law distribution. Formula (1) and (2) consequently show CDF and PDF of power law distribution [1]. To study using real data, we had to focus on one country because

our references presented their result for more than 6 countries. We selected Spain as in [3].

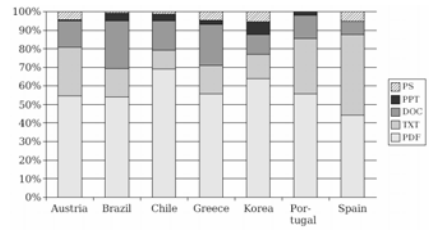
$$P[X > x] \sim x^{-\theta} \quad (1)$$

$$P[X = x] \sim x^{-(\theta+1)} = x^{-a} \quad (2)$$

There are many duplicated URLs which are generated during parsing process of crawling module. To prevent them entering the download phase, we should calculate the probability of duplicated URLs. Each site in Spain has a power law distribution with θ=1.1 for the number of pages (mean=52 pages) [5]. Each HTML object of Spain contains about 15 URLs [5]. According to a power law distribution with θ=X (and mean=7), some of the extracted URLs of a page are located on the same site [5]. It means that there is no need to exchange these URLs by other agents so it saves the time and computations. Fig. 7 shows the details about the stated distributions.



(a) Distribution of server errors

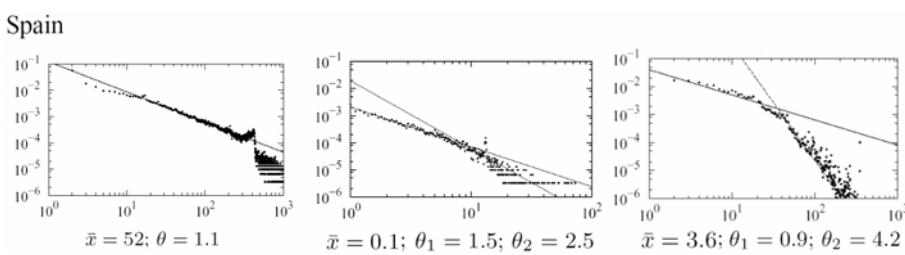


(b) Distribution of MIME objects

File extension	Tool	OK	Conversion	Timeout error	Max. size	Type not allowed	404	Other
.ppt, .pps	xlhtml	19%	54%	1%	15%	1%	9%	2%
.xls	xlhtml	19%	60%	13%	1%	0%	6%	1%
.rtf	unrtf	25%	33%	2%	1%	0%	38%	1%
.swf	webcat	36%	53%	1%	0%	5%	4%	1%
.doc	antiword	54%	33%	2%	1%	0%	8%	2%
.ps	ghostscript	59%	6%	25%	3%	0%	5%	2%
.pdf	xpdf	74%	8%	4%	4%	1%	7%	2%
.txt	-	90%	0%	5%	0%	0%	4%	1%
.html, .htm	webcat	94%	0%	0%	0%	0%	4%	2%

(c) Distribution and details of converting errors for MIME documents

Fig. 6. File distribution and error rates for MIME objects [3, 5]

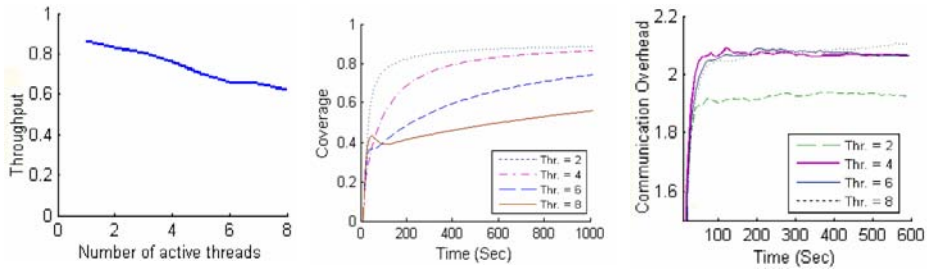


(a) Dist. of page per site (b) Average of internal links (c) Dist. of in-degree of pages

Fig. 7. Distribution of the links and URL's statistics [5]

4 Experimental Results

To evaluate performance of the current model, some measures have been defined in Boldi's work [6]. *Throughput* of a thread is the number of pages which a thread can analyze in a unit of time (*sec*.). We showed that setting up the model using the assumptions of UbiCrawler, the resulting throughput is similar to that of UbiCrawler. Fig. 8 (a) shows throughput diagrams, while the number of threads is increased.



(a) (b) (c)

Fig. 8. Performance measures of agents with different number of threads: (a) throughput, (b) coverage and (c) communication overhead.

Another performance measure which is used here is *coverage*. If *c* is the number of actually crawled pages, and *u* is the number of pages, the crawler as a whole had to visit, then coverage is defined as *c/u* [6]. Fig. 8 (b) shows that by increasing the time (and total number of downloaded URLs) we will reach to a stable *coverage* for the system. This number declares that in steady-state of the system, we will have how many faulty pages. *Communication overhead* is the third measure defined as *e/n*, where *e* is the number of URLs exchanged by the agents during the crawl and *n* is the number of crawled pages [6]. Fig. 8 (c) demonstrates the communication overhead measure if the number of threads is increased.

5 Conclusions

Web crawlers are software systems which can collect files from the Web and extract new URLs to download more objects. According to the size of Internet and update rate of Web sites, a distributed and dependable crawler is needed to download the maximum number of high quality objects. As the task of development of such a crawler is expensive and highly sensitive to the occurrence of faults, there is a need to formally model a crawler before it implementation.

In this paper we introduced a SAN model of a fully-distributed crawler based on UbiCrawler architecture. We used some recently results of Viuva Negra crawler as parameters of the probability distributions used in our model.

The results of the evaluation of SAN model are as follows. Firstly, throughput of the presented model is compared with the existing data from UbiCrawler. Using the model,

we can compute other performance measures to analyze the behavior of the crawler. Secondly, we have evaluated coverage and communication overhead measures.

Future works can include formal definitions of dependability and performability measures of a crawling system. By evaluation of these measures, we can study the effects of different solutions on performability of the system. Modeling a distributed indexing system can also be another future work to construct a fully-distributed dependable search engine.

References

1. Adamic, L.A.: Zipf, Power-Laws, and Pareto - A Ranking Tutorial. White Paper, Information Dynamics Lab, HP Labs, Palo Alto, CA (2000)
2. Avizienis, A., Laprie, C.-J., Randell, B., Landwehr, C.: Basic Concepts and Taxonomy of Dependable and Secure Computing. IEEE Trans. on Dependable And Secure Computing 1(1), 11–33 (2004)
3. Baeza-Yates, R., Castillo, C.: Crawling the Infinite Web. Journal of Web Engineering 6(1), 49–72 (2007)
4. Baeza-Yates, R., Castillo, C., Junqueira, F., Plachouras, V., Silvestri, F.: Challenges on Distributed Web Retrieval. In: Proc. of the 23rd IEEE International Conference on Data Eng. (ICDE 2007) (2007)
5. Baeza-Yates, R., Castillo, C., Efthimiadis, E.N.: Characterization of National Web Domains. ACM Transaction on Internet Technology 7(2) (2005)
6. Boldi, P., Codenotti, B., Santini, M., Vigna, S.: UbiCrawler: a Scalable Fully Distributed Web Crawler. Journal of Software, Practice and Experience 34(8), 711–726 (2004)
7. Brin, S., Page, L.: The Anatomy of a Large-Scale Hyper Textual Web Search Engine. In: Proc. of the 7th International Conference on World Wide Web, pp. 107–117 (1998)
8. Castillo, C.: Effective Web Crawling. PhD Thesis, University of Chile (2004)
9. Cho, J., Garcia-Molina, H.: Parallel Crawlers. In: Proc. of the 11th International Conference on World Wide Web, pp. 124–135. ACM Press, Honolulu (2002)
10. Deavours, D.D., et al.: The Möbius Framework and Its Implementation. IEEE Transaction on Software Engineering 28(10), 956–969 (2002)
11. Exposto, J., Macedo, J., Pina, A., Alves, A., Rufino, J.: Geographical Partition for Distributed Web Crawling. In: Proc. of the 2005 Workshop on Geographic Information Retrieval (GIR 2005), pp. 55–60. ACM Press, New York (2005)
12. Gomes, D., Silva, M.J.: The Viuva Negra Crawler. Technical Report (2006)
13. Heydon, A., Najork, M.: Mercator: A Scalable, Extensible Web Crawler. World Wide Web 2(4), 219–229 (1999)
14. Internet Growth and Statistics: Credits and Background, `http://www.mit.edu/people/mkgray/net/background.html`
15. Kahle, B.: The Internet Archive. RLG Diginews 6(3) (2002)
16. Karger, D., Lehman, E., Leighton, T., Levine, M., Lewin, D., Panigrahy, R.: Consistent Hashing and Random Trees: Distributed Caching Protocols for Relieving Hot Spots on the World Wide Web. In: Proc. of the 29th Annual ACM Symposium on Theory of Computing, El Paso, Texas, pp. 654–663 (1997)
17. Movaghar, A., Meyer, J.F.: Performability Modeling with Stochastic Activity Networks. In: Proc. of the 1984 Real-Time Systems Symposium, Austin, TX, pp. 215–224 (1984)

Performance Comparison of Simple Regular Meshes and Their *k*-ary *n*-cube Variants in Optical Networks

Ahmad Kianrad[1], Aresh Dadlani[1,2], Ali Rajabi[1,2], Mohammadreza Aghajani[3], Ahmad Khonsari[1,2], and Seyed Hasan Seyed Razi[1]

[1] Department of Electrical and Computer Engineering, University of Tehran, Iran
[2] IPM School of Computer Science, Tehran, Iran
[3] Sharif University of Technology, Tehran, Iran
a.kianrad@ece.ut.ac.ir,
{a.dadlani,alirajabi,aghajani,ak}@ipm.ir, seyedraz@ece.ut.ac.ir

Abstract. The need for supporting the ever-growing number of multi-computers in the contemporary Internet has persuaded researches worldwide to search for suitable underlying network topologies with desirable properties. Even in the presence of speedy optical technologies, the arrangement of nodes in a network can be highly influential. Among all the topologies investigated so far, *k*-ary *n*-cubes have been reported to be widely adopted in the literature. But in hybrid networks, where the network is composed of a mixture of different types of topologies, the type and size of *k*-ary *n*-cubes can greatly affect the performance factor of the network. In this paper, we study and compare the performance behaviour of simple regular meshes with their wrap-around variants (2-D and 3-D torus) for Optical Packet Switching (OPS) systems with various sizes under a common traffic condition through simulation results.

Keywords: Optical Packet Switching (OPS), simple regular mesh, *k*-ary *n*-cubes, performance measures.

1 Introduction

In the past few years, networking has been experiencing a migration towards optical-based technologies. The reason for such diversion is the high flexibility, increased scalability, Quality of Service (QoS) management, and unlimited bandwidth provisioning introduced by such optical networks. Among all the proposed paradigms [1-3], OPS has been the subject of several research projects [4, 5]. Its potential to interface with the WDM transport layer and bridge the gap between the electrical (IP) layer and the optical (WDM) layer has added to its popularity [2, 6].

Despite the implementation of OPS as the underlying technology [7, 8], there are yet other prominent factors that influence the performance of networks such as network topology, switching method, routing algorithm and traffic load [9]. The performance of OPS networks with regular and irregular mesh topologies has been studied in the literature [10]. But, due to the high adoption of *k*-ary *n*-cubes in OPS networks, in this paper, we compare the performance factors of the basic instances of such a topology with simple regular meshes for systems with various sizes and under a given traffic load using results obtained through simulations.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 543–550, 2008.

The rest of this paper is structured as follows. In Section 2, we provide an introduction on the types of topologies available and emphasize mainly on *k*-ary *n*-cubes. Next, we delineate the performance factors under investigation in Section 3, followed by the simulation scenarios with relevant explanations on the results obtained through simulations in Section 4. Towards the end, in Section 5, we summarize our work and enlighten directions for possible future works

2 Network Topologies

Network topology is the physical or logical arrangement of nodes interconnected through links. Generally, existing networks are broadly classified into *indirect* and *direct* networks. Examples of *indirect* networks are crossbar and bus, as their nodes are connected to other nodes by means of multiple intermediate stages of switches. On the contrary, in *direct* networks, each node is connected directly to some other nodes, such as in a *k*-ary *n*-cube, mesh and tree. Due to high scalability, *direct* networks, in particular *k*-ary *n*-cubes, have been extensively employed in several networks.

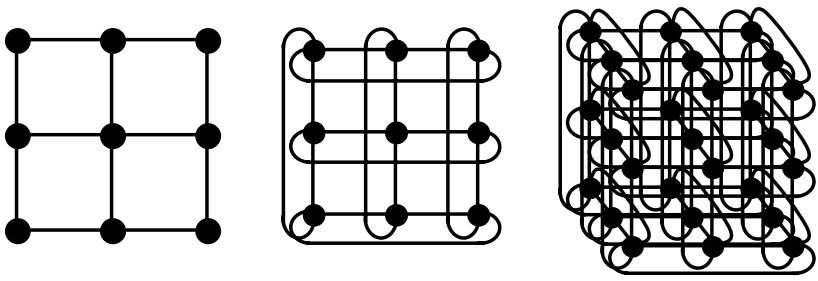


Fig. 1. The various *k*-ary *n*-cubes under consideration (a) a simple 3x3 regular mesh (b) a 3-ary 2-cube (2-D torus), and (c) a 3-ary 3-cube (3-D torus)

The *k*-ary *n*-cube, where *k* denotes the *radix* and *n* symbolizes the *dimension*, is an *n*-dimensional grid structure with *k* nodes accommodated in each dimension. Each of the k^n nodes can be identified by an *n*-digit radix *k* address [11]. Fig. 1 illustrates a simple 3 x 3 regular mesh, followed by instances of *k*-ary *n*-cubes. If we denote the number of nodes in topology i as N_i and the number of edges as L_i, then the following equations hold for regular meshes (SRM) similar to that shown in Fig. 1-a:

$$N_{SRM} = k^n \ , \qquad L_{SRM} = 2k\left(k-1\right) \ . \tag{1}$$

Similarly, for the 2-D and 3-D tori instances of *k*-ary *n*-cubes, as illustrated in Figs. 1-b and 1-c, the number of nodes is the same as that of N_{SRM}, while the number of links are as follows:

$$L_{2-D} = 2k^2 \ , \qquad L_{3-D} = 3k^3 \ . \tag{2}$$

In the following section, we introduce the factors chosen to study the performance of the different instances of each type of topology.

3 Performance Metrics

The four metrics commonly used to evaluate the performance of a network in terms of network utilization, reliability and latency are network throughput, packet-loss rate, average end-to-end delay and mean hop distance [11]. If we denote by P_L and P_T, the total number of packets lost during transmission and the total number of packets generated, respectively, then the packet-loss rate is defined as:

$$Packet-loss\ Rate = \frac{P_L}{P_T} \ . \tag{3}$$

Network throughput is the fraction of network resource that successfully delivers data. Because packets are dropped, a part of the network capacity is wasted in transporting the bits that are dropped. An ideal situation in which the network throughput tends to reach unity is when no packets are dropped and no link remains idle. Thus, the network throughput can be defined as:

$$Network\ Throughput = \frac{N_{bits}}{\left(\frac{C_{network} \times T_{simulation}}{H_{ideal}} \right)} \ . \tag{4}$$

where N_{bits}, $C_{network}$, $T_{simulation}$ and H_{ideal} denote respectively, the total number of bits successfully delivered, the transmission capacity of the network, simulation time and the ideal average hop distance. In (4), $C_{network}$ is, in turn, calculated as follows:

$$C_{network} = N_{link} \times N_{wavelength} \times R_{data} \ . \tag{5}$$

in which N_{link}, $N_{wavelength}$ and R_{data} symbolize respectively, the total number of links, total number of wavelengths and data rate. Mean hop distance is the number of hops taken by a packet to traverse a path, averaged over all possible source-destination pairs in the network.

Since our main interest is to study the topologies feasible for the optical domain, there exist other metrics that should be regarded in a network. The primary key metric is the *cost* of fibers required to implement a network with higher performance. Due to the high cost of optical fibers, it has always been feasible to minimize the amount of fiber used in optical networks, unless the addition of a few links would efficiently improve the performance. This metric is in trade-off with the average number of hops mentioned earlier and depends on the structure of the topology under study. It is upon the routing algorithms to decide whether to utilize an extra fiber link or increase the

hop distance of a packet traveling through the network. Based on the cost and mean hop distance, we account another metric called the *routed capacity*. This factor determines the percentage of packets routed through the network by the routing algorithm in use. Another metric dependent on the trade-off between fiber cost and mean hop distance is *link utilization* which indicates the percentage of available links in use. In order to study the performance behavior of different *k*-ary *n*-cubes with different sizes under the optical domain, it is mandatory to analyze such systems through predefined or meaningfully derived measures.

4 Simulation Scenarios and Results

For purpose of illustration, the network topologies under study are size variants of those shown in Fig. 1. For sake of clarity, we use the notation $T_k(i)$ to denote a topology of type k, $(k \in \{SRM, 2-D, 3-D\})$, and size i, $(i \in \{4,6,8,10,12,14,16\})$. For instance, a simple 6x6 regular mesh is denoted as $T_{SRM}(6)$, a 10-ary 2-cube by $T_{2-D}(10)$, and a 4-ary 3-cube by $T_{3-D}(4)$.

We have created and compared our scenarios in the OPNET WDM Guru environment. In each of the following scenarios, each bi-directional link i is L_i long. The number of fiber pairs in each link is N_f and the user-defined cost of each pair is C_f. Every fiber contains W wavelengths, each carrying a data stream at rate R. In addition, the traffic matrix is generated randomly, and all packets are chosen to be routed through the shortest path. The values for the parameters used in the simulation are L_i = 20 km, N_f = 50, and C_f = 50.

In our first scenario, we compare the three aforementioned topologies on the basis of their routed capacity. As shown in Fig. 2, it is obvious that as *k*-ary 3-cubes are more effective in routing packets than *k*-ary *2*-cube and simple regular mesh for smaller sized topologies, this percentage decreases with increase in topology size. This proves the fact that in spite of the being very expensive, *k*-ary 3-cubes provide better performance than the other topologies for smaller-sized networks, but as the size increases, it is not feasible to adopt such *k*-ary 3-cubes as their routing capacity approaches that of simple regular meshes.

In Fig. 3, the total hop count in *k*-ary 3-cubes is far greater than that in case of simple regular meshes and *k*-ary 2-cubes. In fact, the total hop counts of the latter two nearly overlap with each other. This implies that the cost of taking more number of hops is far lesser than that of increasing the fiber cost value. Thus, routing algorithms that consider fiber and hop counts as their routing cost parameter tend to increase the hop distance rather than add to the fiber cost.

Our next comparison involves the mean hop count for topologies with various sizes. As illustrated in Fig. 4, it can be easily deduced that the expected mean hop count would have a trend similar to that of the total hop count depicted in Fig. 3. As mean hop is the overall average of the total hop count taken to travel all possible source-destination pairs in the network, it can be inferred that in *k*-ary 3-cubes, the fiber cost is minimized at the expense of the mean hop value which, in turn, depends on the total number of hops.

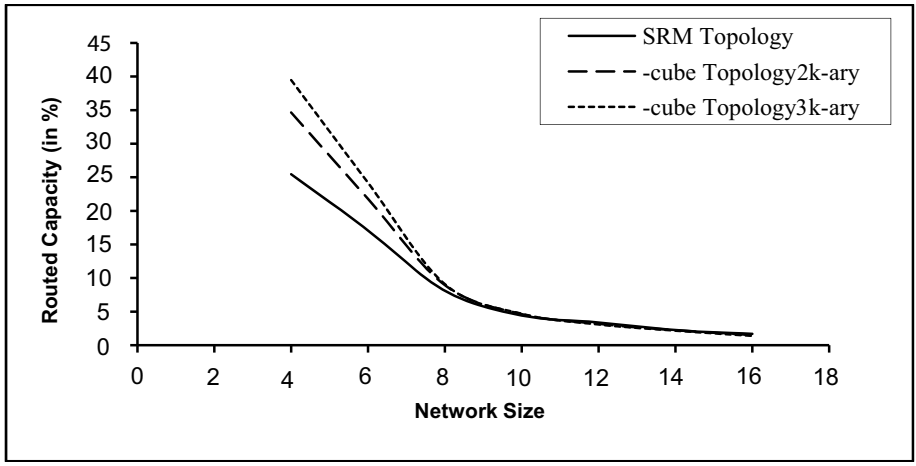


Fig. 2. Performance comparison of simple regular meshes, *k*-ary 2-cubes, and *k*-ary 3-cubes in terms of routed capacity and under random traffic condition. The sizes of all three topologies are even numbers ranging from 4 to 16.

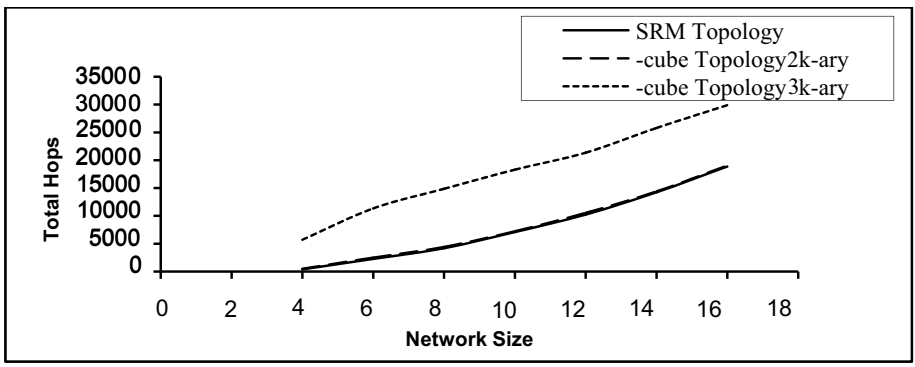


Fig. 3. Performance comparison of simple regular meshes, *k*-ary 2-cubes, and *k*-ary 3-cubes in terms of total number of hops and under random traffic condition. The sizes of all three topologies are even numbers ranging from 4 to 16.

The final measurable metric is the network link utilization. Keeping in mind the random nature of traffic and the policies implemented by routing algorithms, the trade-off between fiber cost and hop count can be clearly seen in Fig. 5. Comparing Fig. 4 and 5, we can conclude that the routing cost (fiber cost + hop count) of *k*-ary 3-cubes is greater than that of *k*-ary 2-cubes, which in turn, is greater than that of simple regular meshes. At the beginning, the fiber link utilization percentage of the three topologies is almost the same, but then diverges as the network size increases.

In order to study the relation between routed capacity and link utilization, we defined the maximum number of hops to be some constant value. As can be inferred

from Table 1, with increase in network complexity, the routing cost decreases. That is, it is not quite feasible to adopt topologies of large sizes in order to enhance the network performance. Also, for topologies with smaller sizes, routing algorithms tend to perform well in *k*-ary 3-cubes than their simple mesh and *k*-ary 2-cube variants.

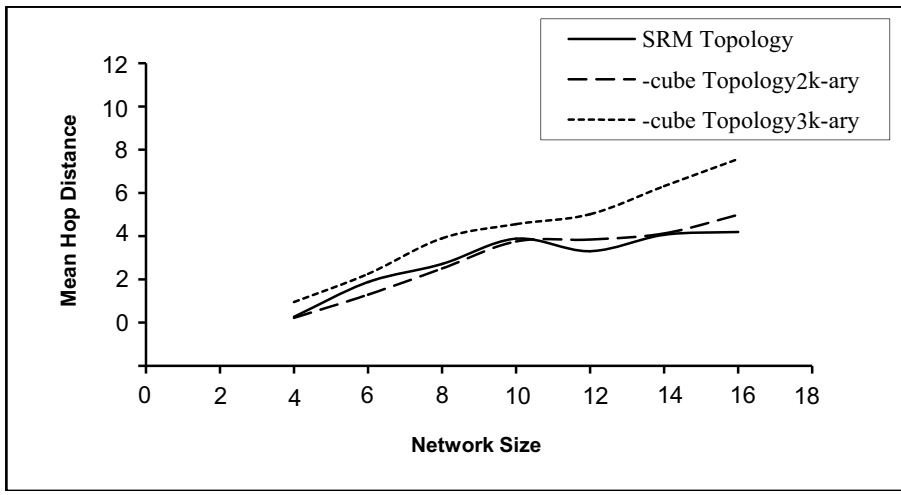


Fig. 4. Performance comparison of simple regular meshes, *k*-ary 2-cubes, and *k*-ary 3-cubes in terms of mean hop distance traveled by a packet and under random traffic condition. The sizes of all three topologies are even numbers ranging from 4 to 16.

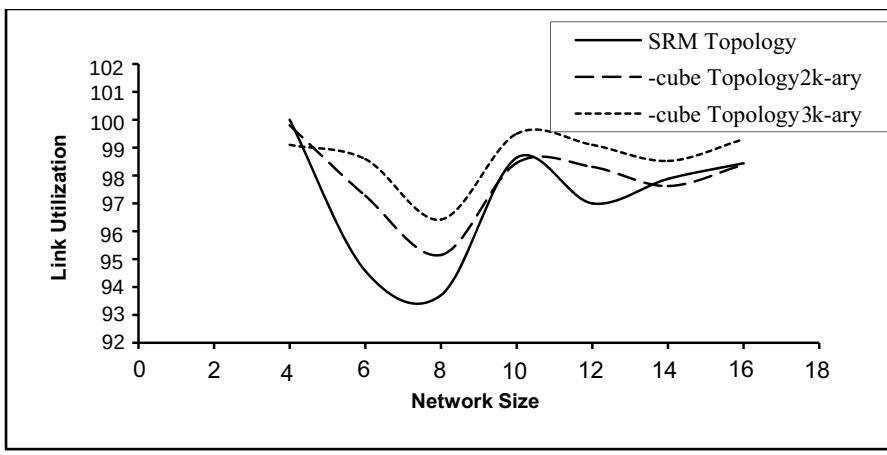


Fig. 5. Performance comparison of simple regular meshes, *k*-ary 2-cubes, and *k*-ary 3-cubes in terms of link utilization and under random traffic condition. The sizes of all three topologies are even numbers ranging from 4 to 16.

Table 1. Comparison results obtained through simulations for all performance metrics. The four performance measures are listed as column labels and the size and types of topologies under consideration are listed as row labels. SRM, 2-D and 3-D denote, respectively, a simple regular mesh, a *k*-ary 2-cube, and a *k*-ary 3-cube.

		Routed Capacity (%)	Total Hops	Mean Hop	Link Utilization (%)
4	SRM	25.45	384	2.27	100
	2-D	34.64	511	2.22	99.8
	3-D	39.45	5708	2.95	99.1
6	SRM	17.09	2270	3.87	94.58
	2-D	21.78	2490	3.29	97.27
	3-D	24.18	11357	4.25	98.59
8	SRM	8.11	4198	4.71	93.71
	2-D	8.94	4415	4.5	95.15
	3-D	9.1	14871	5.9	96.42
10	SRM	4.44	7102	5.88	98.64
	2-D	4.63	7246	5.75	98.45
	3-D	4.69	18324	6.55	99.5
12	SRM	3.4	10243	5.3	97
	2-D	3.18	10539	5.84	98.31
	3-D	3.1	21359	7.01	99.1
14	SRM	2.26	14252	6.06	97.88
	2-D	2.23	14408	6.12	97.62
	3-D	2.18	25812	8.31	98.52
16	SRM	1.7	18861	6.19	98.44
	2-D	1.5	19048	6.98	98.39
	3-D	1.2	29903	9.57	99.3

5 Conclusions

Performance measures adopted to study network behavior with various underlying topologies have been plenty. In this paper, we studied and analyzed the performance behavior of the three most efficient topologies namely, simple regular meshes, *k*-ary 2-cubes and *k*-ary 3-cubes in terms of routed capacity, mean hop distance and network link utilization. We learnt that routing algorithms with routing cost defined as fiber cost and hop count, tend to boost network performance in case of smaller sized topologies. But with increase in network size, the performance gradually reduces independent of the type of topology chosen. Thus, it is not feasible to adopt large sized *k*-ary *n*-cubes in any hybrid network as such topologies would not only increase the cost of implementation, but also perhaps increase performance insignificantly. The work done in this paper can be further extended to involve other feasible topologies and k-ary *n*-cube variants studied under real-world traffics existing in the optical network.

References

1. Meagher, B.: Design and Implementation of Ultra-low Latency Optical Label Switching for Packet-switched WDM Networks. IEEE/OSA Journal of Lightwave Technology 18, 1978–1987 (2000)
2. O'Mahony, M.J., Simeonidou, D., Hunter, D.K., Tzanakaki, A.: The Application of Optical Packet Switching in future Communication Networks. IEEE Communications Magazine 39, 128–135 (2001)
3. Qiao, C., Yoo, M.: Optical Burst Switching (OBS) - A new Paradigm for an Optical Internet. Journal of High Speed Networks 8(1), 69–84 (1999)
4. Gambini, P.: Transparent Optical Packet Switching: Network Architecture and Demonstrators in the KEOPS Project. IEEE Journal on Selected Areas in Communications 16(2), 1245–1259
5. Dittmann, L.: The European IST Project DAVID: A viable Approach towards Optical Packet Switching. IEEE Journal on Selected Areas in Communications 21(7), 1026–1040 (2003)
6. Yao, S., Yoo, S.J.B., Mukherjee, B., Dixit, S.: All-optical Packet Switching for Metropolitan Area Networks: Opportunities and Challenges. IEEE Communications Magazine 39, 142–148 (2001)
7. Callegati, F., Cankaya, A.C., Xiong, Y., Vandenhoute, M.: Design Issues of Optical IP Routers for Internet Backbone Applications. IEEE Communications Magazine 37, 124–128 (1999)
8. Callegati, F., Cerroni, W.: Wavelength Allocation Algorithms in Optical Buffers. In: Proceeding of IEEE International Conference on Communications (ICC 2001), vol. 2, pp. 499–503 (2001)
9. Duato, J., Yalamanchili, S., Ni, L.M.: Interconnection Networks: An Engineering Approach. IEEE Computer Science Press, Los Alamitos (1997)
10. Mukherjee, B.: Optical WDM Networks. Springer, Heidelberg (2006)
11. Min, G.: Performance Modelling and Analysis of Multi-computer Interconnection Networks (January 2003)

A Robust and Efficient SIP Authentication Scheme*

Alireza Mohammadi-nodooshan[1], Yousef Darmani[1], Rasool Jalili[2], and Mehrdad Nourani[3]

[1] Electrical Engineering Department, K.N.Tossi University of Technology, Tehran, Iran
{mohammadi@ee,darmani@eetd}.kntu.ac.ir
[2] Computer Engineering Department, Sharif University of Technology, Tehran, Iran
jalili@sharif.edu
[3] Electrical Engineering Department, The University of Texas at Dallas, TX, USA
nourani@utdallas.edu

Abstract. Apart from its security, an SIP (Session Initiation Protocol) authentication protocol shall be efficient; because in order to replace traditional telephony, VoIP services have to offer enough security and QoS compared to PSTN services. Recently, Srinivasan et al. proposed an efficient SIP authentication scheme. The low delay overhead introduced by their scheme, causes the total call setup time to be well within the acceptable limit recommended by ITU-T. Based on their work, this paper proposes an SIP authentication scheme. In both schemes, the end users are authenticated with the proxy server in their domain using the registrar server. Comparing with the Srinivasan et al.'s scheme, our proposed scheme is more efficient and secure. Our scheme's low overhead makes it suitable for applying as an authentication protocol in SIP networks.

1 Introduction

SIP is becoming the de facto signaling standard for the next-generation VoIP networks. SIP is a request-response protocol. In an SIP session, A User agent Client (UC) initiates a request and a User agent Server (US) responses to the request. The Proxy Server (PS) receives a request and forwards it to the user agent server; it also forwards the response received from the user agent server to the user agent client. When the user agent client asks the proxy server to forward its request, it has to be authenticated to the proxy server. However, a major problem with SIP security is lack of a strong authentication mechanism. The main SIP authentication mechanism addressed in the SIP RFC is based on the HTTP digest authentication; although, as noted in [1], it does not provide a very high level of security.

One of the main requirements of any authentication protocol is resisting against replaying and forging attacks. It should also require no password table at the server side. This is due to several disadvantages of maintaining a password table. Some two-party ID-based schemes like [2] and [3] were proposed to fulfill the aforementioned requirements. However, authentication in SIP involves three parties and the user agent

* This work is partially supported by Iran Telecommunication Research Center (ITRC) under grant number 500/9119.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 551–558, 2008.

client needs to authenticate itself to the proxy server, whereas the registration information of the user client is stored in the registrar server (RS). To prevent from man-in-the-middle attacks, the authentication also has to be mutual. The authentication mechanism shall also establish a session key between the communicating agents that can be used by the encryption process between them.

Recently, Srinivasan et al. [4] mentioned an SIP authentication protocol that fulfills all the above requirements and its computation overhead is only 10 milliseconds[1]. As they discuss, this low overhead causes the total call setup time to be well within the acceptable limit that is recommended by ITU-T [4]. Their scheme uses hash functions, symmetric, private key and public key encryption operations. Due to the huge computation burden of private key operations, this paper proposes a more secure SIP authentication scheme, using less private key operations. For prevention of DoS attacks, the proposed scheme also moves the unnecessary computation load on the SIP servers to the SIP clients.

This paper is organized as follows: Next section reviews the Srinivasan et al.'s scheme. Section 3 proposes our authentication scheme. Sections 4 and 5 analyze efficiency and security of the proposed scheme. Section 6 concludes the paper.

2 Review of Srinivasan et al.'s Scheme

The notations used throughout this paper are summarized in Table 1.

In the Srinivasan et al.'s scheme, first, UC registers in RS. During this phase, UC submits I_{UC} to RS and RS generates PW_{UC} and r values and sends them to UC where:

$$PW_{UC} = \mathrm{H}[N \,||\, I_{UC}] \tag{1}$$

$$r = \mathrm{H}[N \,||\, I_{RS}] \oplus \mathrm{H}[N \,||\, I_{UC}] \oplus I_{RS} \oplus I_{UC} \tag{2}$$

N is a large number and is kept secret by RS. (1) and (2) are executed in T_H[2].

The second phase is started as UC wants to establish a connection with US and needs to be authenticated to PS. This phase is completed in the following four steps:

Step1. UC generates the random number R_0 and $\mathrm{UC} \rightarrow \mathrm{PS} : A$

$$A = n,\ (R_0)_L,\ I_{RS},\ TS_{UC} \tag{3}$$

$$n = r \oplus PW_{UC} \tag{4}$$

$$L = \mathrm{H}[PW_{UC} \oplus TS_{UC}] \tag{5}$$

[1] With P III 1.0 GHz Processors.

[2] $\mathrm{H}[N||I_{RS}]$ is precomputed in RS. The overhead of the XOR operation is negligible.

Table 1. Notations

I_e	identifier of an entity 'e'
TS_e	time stamp generated by an entity 'e'
$(M)_K$	message 'M' encrypted using a symmetric key 'K'
$E_K(M)$	message 'M' encrypted using an asymmetric key 'K'
KR_e	private key of an entity 'e'
KU_e	public key of an entity 'e'
C_e	certificate of an entity 'e'
H[M]	output digest of a hash function 'H' with 'M' as its input
$e1 \rightarrow e2 : V$	entity 'e1' sends vector 'V' to entity 'e2' via an insecure channel
$e1 \Rightarrow e2 : V$	entity 'e1' sends vector 'V' to entity 'e2' via a secure channel
T_H	average execution time of a hash function
T_S	average execution time of a symmetric encryption or decryption
T_{PR}	average execution time of an operation which uses a KR_e for encryption or decryption (e.g. signing)
T_{PU}	average execution time of an operation which uses a KU_e

Step1 imposes a computation overhead of $T_H + T_S$ on UC.

Step2. PS checks TS_{UC} for its freshness, generates a secret random number σ and $PS \rightarrow RS : B$.

$$B = \sigma,\ n,\ (R_0)_L,\ TS_{UC},\ Signature\ of\ PS,\ TS_{PS},\ C_{PS} \tag{6}$$

$$Signature\ of\ PS = E_{KR_{PS}}(H[\sigma,\ n,\ (R_0)_L,\ TS_{UC},\ C_{PS}]) \tag{7}$$

This step imposes an overhead of $T_{PR}+T_H$ on PS.

Step3. RS validates TS_{PS}, C_{PS} and *Signature of PS*. Then, RS finds I_{UC} by: $I_{UC} = I_{RS} \oplus n \oplus H[N \,||\, I_{RS}]$ and checks whether UC is a legal user. If so, RS computes L to decipher $(R_0)_L$. Then, RS ciphers $E_{KU_{PS}}(H[I_{UC}] \,||\, R_0)$ and generates:

$$Signature\ of\ RS = E_{KR_{RS}}(H[\sigma,\ \gamma,\ E_{KU_{PS}}(H[I_{UC}] \,||\, R_0)]) \tag{8}$$

γ is a secret random number generated by RS. Then: $RS \rightarrow PS : C$

$$C = \gamma,\ E_{KU_{PS}}(H[I_{UC}] \,||\, R_0),\ Signature\ of\ RS,\ TS_{RS},\ C_{RS} \tag{9}$$

This step needs $T_{PR}+3T_{PU}+T_S+6T_H$[3] of processing time.

Step4. PS verifies validity of TS_{RS}, C_{RS} and *Signature of RS*. If they are valid, PS identifies that UC is an authorized user and issues a temporary certificate (TC_{UC}) for UC. Using its private key, PS then decrypts $E_{KUPS}(H[I_{UC}] \,||\, R_0)$ and computes the session key:

$$SK = H[I_{UC}] \oplus R_0 \tag{10}$$

[3] In this paper, we assume using the ITU-T recommendation X.509 with a CA hierarchy height of one level; Therefore, validating (the signature part of) C_{PS} is completed in only T_H+T_{PU}. Note that, validating *Signature of PS* takes $T_{PU}+T_H$ of processing time.

Then it sends $(TC_{UC})_{SK}$ to UC. *Step4* is completed in $2T_{PR}+2T_{PU}+T_S+3T_H$[4].

Now UC computes *SK* and obtains TC_{UC} which introduces a processing time of T_S[5]. *SK* is also used for encryption of further signaling between PS and UC.

After this step, through a mechanism mentioned in Srinivasan et al.'s scheme and using the R_0 and TC_{UC} values, UC authenticates itself to US periodically.

3 The Proposed Scheme

Our proposed scheme includes the following three protocols:

3.1 The Registration Protocol (RP)

In this protocol, the following steps are performed:

RP1) $\mathrm{UC} \Rightarrow \mathrm{RS} : ID_{UC}$

RP2) $\mathrm{RS} \Rightarrow \mathrm{UC} : PW_{UC}, C_{RS}$. PW_{UC} is computed the same as in (1).

3.2 The Login Protocol (LP)

This Protocol has only one step:

LP1) $\mathrm{UC} \rightarrow \mathrm{PS} : A'$

$$A' = I_{UC},\ authToken, TS_{UC},\ C_{RS} \tag{11}$$

$$authToken = \mathrm{H}[TS_{UC} \,\|\, PW_{UC}] \tag{12}$$

3.3 The Authentication Protocol (AP)

This protocol authenticates UC and sets a session key between UC and PS (SK_{UP}), besides a session key between RS and PS (SK_{RP}). The AP is completed in four phases:

AP-Phase 1. Upon receiving *A'*, PS performs the following steps:

AP1) PS verifies freshness of TS_{UC}.

AP2) PS validates C_{RS}[6] and gets KU_{RS} out of C_{RS}. This step is completed in $T_{PU}+T_H$.

AP3) PS generates a random number SK_{RP} and computes the value:

$$encSK_{RP} = \mathrm{E}_{KU_{RS}}(TS_{UC} \,\|\, SK_{RP}) \tag{13}$$

AP4) PS computes

[4] Issuing TC_{UC} takes T_H+T_{PR} of processing time.

[5] Consider that $H[I_{UC}]$ is precomputed in UC.

[6] Note that, version 3 of the X.509 specification is used in our scheme and the certificate issuer utilizes the "Subject" and "Extensions" elements of the certificate to declare the roles which the holder of the certificate can act out in an SIP network; therefore, PS validates C_{RS} to check whether this certificate belongs to a valid registrar server or not.

$$SK_{UP} = \mathrm{H}[SK_{RP} \,||\, TS_{UC}] \tag{14}$$

and saves both, SK_{UP} and SK_{RP}, for the current session.

AP5) PS → RS : *B'*

$$\mathrm{B'} = I_{UC}, authToken, encSK_{RP} \tag{15}$$

AP-Phase 2. Upon receiving *B'*, RS performs the following steps:

AP6) RS verifies whether UC is a legal user.

AP7) RS decrypts $encSK_{RP}$.

AP8) RS checks whether TS_{UC} is within some elapsed time Δt (TS_{RS} - $TS_{UC} < \Delta t$).

AP9) RS checks if $\mathrm{H}[TS_{UC} \,||\, \mathrm{H}[N \,||I_{UC}]] = authToken$. This step needs $2\mathrm{T_H}$.

AP10) RS computes SK_{UP} the same as in (14) and saves SK_{RP} for this session.

AP11) RS masks SK_{UP} with a "one-time common secret between RS and UC" to compute $maskedSK_{UP}$ as in (16). This step is completed in $\mathrm{T_H}$.

$$maskedSK_{UP} = SK_{UP} \oplus \mathrm{H}[PW_{UC} \,||\, TS_{UC}] \tag{16}$$

AP12) RS computes

$$RSMac = \mathrm{H}[maskedSK_{UP} \,||\, SK_{UP}] \tag{17}$$

AP13) RS → PS *: C'*

$$C' = maskedSK_{UP}, RSMac \tag{18}$$

AP-Phase 3. Upon receiving C', PS performs the following steps:

AP14) Using the saved value of SK_{UP}, PS verifies whether $\mathrm{H}[maskedSK_{UP} \,||SK_{UP}] = RSMac$. If this verification is passed, PS discovers the authenticity of the UC's request. This step is completed in $\mathrm{T_H}$.

AP15) Using its private key, PS issues a temporary certificate for UC. This step needs $\mathrm{T_{PR}}+\mathrm{T_H}$ of processing time.

AP16) PS computes $(TC_{UC})_{SK_{UP}}$.

AP17) PS → UC : *D'*

$$D' = maskedSK_{UP}\ ,\ (TC_{UC})_{SK_{UP}}\ ,\ C_{PS} \tag{19}$$

AP-Phase 4. Upon receiving D', UC performs the following steps:

AP18) UC calculates $\mathrm{H}[PW_{UC} \,||\, TS_{UC}]$ to compute SK_{UP} as: $SK_{UP} = maskedSK_{UP} \oplus \mathrm{H}[PW_{UC} \,||\, TS_{UC}]$.

AP19) Using SK_{UP} , UC decrypts $(TC_{UC})_{SK_{UP}}$.

AP20) UC validates C_{PS} and checks whether it is a valid proxy server certificate. Then, UC gets KU_{PS} out of C_{PS}. Using KU_{PS}, PS validates the sign part of TC_{UC} and if it is valid, UC saves TC_{UC} and SK_{UP} for the current session. Otherwise, UC finds out that PS is a bogus proxy server and will terminate any transactions with this proxy

server. UC will also inform RS about this forgery. AP20 needs $2T_{PU}+2T_H$ of processing time. At this point:

1 - Both schemes (Srinivasan et al.'s and ours) have authenticated UC to PS and issed a temporary certificate for UC.

2 - Both schemes have set a session key between PS and UC.

During the conversation, voice of the end users acts as an authentication factor between them. However, if a strict authentication is needed between the end users; the procedure used in the Srinivasan et al.'s scheme for authenticating the end users to each other is utilized in our scheme too. However, before running this procedure, the value of R_0 shall be shared between UC and PS. In our scheme, PS and UC will compute the value of $H[SK_{UP}||TS_{UC}]$ and use this value as R_0. It is a random value and a common secret between PS and UC.

4 Performance Analysis

As the proposed scheme is based on the Srinivasan et al.'s work, this section reviews the computation and communication performance of our scheme in comparison with Srinivasan et al.'s scheme.

In the registration phase, both schemes introduce the same computation overhead of T_H. Table 2 compares "the computation overhead of LP and AP on each of the SIP entities" in our scheme and in Srinivasan et al.'s scheme. Note that for typical encryption algorithms: $T_{PR} > T_{PU} >>> T_S > T_H$ [5].

Table 2. Comparison of the computation overhead on each of the SIP entities[A]

Scheme	User Client	Proxy Server	Registrar Server
Srinivasan et al.	$2T_S+T_H$	$3T_{PR}+2T_{PU}+T_S+4T_H$	$T_{PR}+3T_{PU}+T_S+6T_H$
Our scheme	$2T_{PU}+T_S+4T_H$	$T_{PR}+2T_{PU}+T_S+4T_H$	$T_{PR}+5T_H$

A: The two additional hash operations required for computing the value of R_0 ($H[SK_{UP}||TS_{UC}]$) in PS and UC in our scheme are also considered in the above table.

From table 2, our scheme employs less private key encryption/decryption operations. This causes the proposed scheme to entail a low computation overhead. Using the Crypto++ 5.5 library and applying RSA-1024, AES/ECB-256 and SHA-256 as the encryption algorithms under Windows XP SP2 in 32-bit mode with Intel Core II 1.83 GHz processors utilized in UC, PS and RS, an average computation overhead of 4 milliseconds[7] is introduced by the proposed scheme.

Introducing a low communication overhead to the VoIP network is one of the other main criteria of any protocol utilized in VoIP environments. It can simply be verified that employing the same encryption functions, the total traffic overload in the VoIP network (introduced for exchanging all of the vectors *A', B', C'* and *D')* in our scheme is less than Srinivasan et al.'s scheme.

[7] This result is valid where only one core of the processors is used at each moment.

5 Security Analysis and Discussions

This section discusses some security points of the proposed scheme.

P1. In the registration protocol, in our scheme, RS does not issue the *r* value for UC and UC sends its identity (I_{UC}) to PS in plaintext in the login protocol. In Srinivasan et al.'s scheme, the *r* value is used to compute *n*. To prevent from sending I_{UC} in clear format, the *n* value is sent to PS. We argue that it is easy for an eavesdropper to derive I_{UC} from *n*. To perform this, the eavesdropper *"Eve"* registers with RS and gets its own r_{Eve} ,$PW_{Eve.}$ Then, *Eve* does the following computation:

$$I_{UC} = n \oplus r_{Eve} \oplus I_{Eve} \oplus PW_{Eve} \tag{20}$$

Therefore, sending *n* is as secure as sending I_{UC} in plaintext and using *r* and *n* only imposes some extra overhead on the authentication scheme.

P2. Influence of TS_{UC} and the random number SK_{RP} in most of the terms used in the authentication scheme, makes future forgery or replay attacks practically unfeasible.

P3. AP1 makes replaying of A' unfeasible. AP2 and AP9 make forging of A' unfeasible. AP14 makes forging of C' impossible.

P4. UC is authenticated to RS at AP9. After passing this step successfully, UC is authorized to authenticate PS at AP20.

P5. RS is authenticated to PS at AP14; because only RS is able to decrypt $encSK_{RP}$ and calculate *RSMac*.

P6. RS is authenticated to UC at AP20; because the value of $H[PW_{UC}||TS_{UC}]$ which is used to unmask $maskedSK_{UP}$, is a common secret between RS and UC.

P7. To prevent from man-in-the-middle attacks, after a normal running of the protocol, the authentication between all of the SIP entities becomes mutual.

P8. Knowing the value of SK_{UP}, UC cannot find SK_{RP}.

P9. Knowing the value of R_0, US cannot find SK_{UP}. In Srinivasan et al.'s scheme, since the values I_{UC} and R_0 are accessible to US, US can compute the session key between UC and PS (SK).

P10. AP8 is designed to prevent an attacker from "impersonating PS[8] to perform any future attacks[9]". However, in some cases AP8 has to tolerate a small delay ($\Delta t'$) to receive B'; because Δt is influenced by not only the network transmission delay, but also the delay caused by the current computation overhead on the legitimate PS of the domain, which is not exactly predictable. In these cases, an attacker, with a high computation power, can exploit the tolerance to pass AP8; but AP9, AP14 or AP20 will stop him/her to perform any attacks using this exploitation. However, this will impose some overheads on UC, PS and RS to be detected. By enumerating these states, it can be verified that in the worst cases, overheads of $T_{PR}+2T_{PU}+4T_H$ or $T_{PR}+2T_{PU}+T_S+8T_H$ are imposed on PS, RS and UC to detect these situations.

P11. As mentioned in [1], SIP is especially prone to DoS attacks. One way to perform this attack is to flood an SIP server with forged messages (which will exhaust a server's processor to be discovered). Due to this consideration, the proposed

[8] To perform this impersonation, the attacker may intrude on the communication line between an UC and PS to obtain a valid A'. Using the valid A', she/he then, computes a forged B' and sends the forged B' to RS. She/he may also replay an eavesdropped valid B' to RS.

[9] e.g. eavesdropping the UC's conversations.

authentication scheme is designed to detect the forged messages with the least possible computation overhead on the SIP servers. To demonstrate this, next point compares "the overhead imposed on the SIP entities to detect forged messages in the worst case", in our scheme and in the Srinivasan et al.'s scheme.

P12. In the Srinivasan et al.'s scheme, an eavesdropper may dispatch an eavesdropped valid *A* vector with an updated value of TS_{UC} to PS. This will not be detected before reception of *D* by UC and imposes an overhead of $4T_{PR}+5T_{PU}+2T_S+10T_H$ on PS and RS. As noted in P10, in the worst cases, in our scheme, overheads of $T_{PR}+2T_{PU}+4T_H$ or $T_{PR}+2T_{PU}+T_S+8T_H$ are imposed on PS, RS and UC to detect forged messages.

6 Conclusion

We demonstrated the demand for a robust and lightweight user authentication scheme in SIP networks. Then, the authentication requirements in SIP were noted. We also reviewed the Srinivasan et al.'s scheme which is one of the most efficient SIP authentication schemes (with an overhead within the IETF standard) proposed. Then, based on the Srinivasan et al.'s scheme, we proposed an SIP authentication scheme which has the following advantages to their scheme:

1. It enhances the security of their scheme and is more resistant to DoS attacks.
2. It is more efficient in terms of both communication and computation cost and reduces the workload on SIP servers.

Under the mentioned conditions, a computation overhead of 4 milliseconds is imposed by the proposed scheme. This low overhead makes the scheme suitable to be applied as an SIP authentication protocol in VoIP networks.

References

1. Salsano, S., Veltri, L., Papalilo, D.: SIP Security Issues: The SIP Authentication Procedure and Its Processing Load. IEEE Network 16(6), 38–44 (2002)
2. Sun, H.M.: An Efficient Remote User Authentication Scheme Using Smart Cards. IEEE Transactions on Consumer Electronics 46(4), 958–961 (2000)
3. Chien, H.Y., Jan, J.K., Tseng, Y.M.: An Efficient and Practical Solution to Remote Authentication: Smart Card. Computers & Security 21(4), 372–375 (2002)
4. Srinivasan, R., et al.: Authentication of Signaling in VoIP Applications. In: 11th Asia-Pacific Conference on Communications (APCC 2005) Proceeding, pp. 530–533 (2005)
5. Menascé, D.: Security Performance. IEEE Internet Computing 7(3), 84–87 (2003)
6. (August 27, 2007), `http://www.cryptopp.com`
7. Sisalem, D., Kuthan, J., Ehlert, S.: Denial of Service Attacks Targeting a SIP VoIP Infrastructure: Attack Scenarios and Prevention Mechanisms. IEEE Network 20(5), 26–31 (2006)
8. Kwang, K., Choo, R., Boyd, C., Hitchcock, Y.: On Session Key Construction in Provably Secure Protocols. In: Dawson, E., Vaudenay, S. (eds.) Mycrypt 2005. LNCS, vol. 3715, pp. 116–131. Springer, Heidelberg (2005)
9. Rosenberg, J., et al.: SIP: Session Initiation Protocol. IETF RFC 3261((2002)

A Temporal Semantic-Based Access Control Model

Ali Noorollahi Ravari, Morteza Amini, and Rasool Jalili

Network Security Center, Computer Engineering Department,
Sharif University of Technology, Tehran, Iran
noorollahi@ce.sharif.edu, m_amini@ce.sharif.edu,
jalili@sharif.edu

Abstract. With the advent of semantic technology, access control cannot be done in a safe way unless the access decision takes into account the semantic relationships between entities in a semantic-aware environment. SBAC model considers this issue in the decision making process. However, time plays a crucial role in new computing environments which is not supported in this model. In this paper we introduce temporal semantic based access control model (TSBAC), as an extension of SBAC model, which enhances the specification of user-defined authorization rules by constraining time interval and temporal expression over users' history of accesses. A formal semantics for temporal authorizations is provided and conflicting situations (due to the semantic relations of the SBAC model and a sub-interval relation between authorizations) are investigated and resolved in our proposed model.

Keywords: Access control, semantic-awareness, temporal authorization, access history.

1 Introduction

Access control is a mechanism that allows owners of resources to define, manage and enforce access conditions applicable to each resource [1]. An important requirement, common to many applications, is related to the temporal dimension of access permissions. In these systems, permissions are granted based on previous authorizations given to the users of the system in specific time points.

Another critical requirement is the possibility of expressing the semantic relationships that usually exist among different authorization elements, i.e. subjects, objects, and actions. To overcome this challenge, our model is constructed based on SBAC model [2] which is a semantic-based access control model. SBAC authorizes users based on the credentials they offer when requesting an access right. Ontologies are used for modeling entities along with their semantic interrelations in three domains of access control, namely subjects domain, objects domain, and actions domain. To facilitate the propagation of policies in these three domains, different semantic interrelations can be reduced to the subsumption relation.

In this paper we propose an access control model characterized by temporal authorizations and based on SBAC model. In the proposed model, a temporal expression is associated with each authorization, identifying the instants in which the authorization applies. Furthermore, a temporal interval bounds the scope of the temporal

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 559–568, 2008.

expressions (e.g., $[1,16]$ shows that the authorization is valid for time interval starting at '1' and ending at '16'). Thus, the main feature provided by our model is the possibility of specifying authorization rules which express temporal dependencies among authorizations. These rules allow derivation of new authorizations based on the presence or absence of other authorizations in specific past time instants (stored in *History Base* in the form of a "time point" and an authorization tuple $(s,o,\pm a)$). By using authorization rules, many protection requirements can be specified. For example, to specify that a user has an authorization as long as another user has this authorization; or that a user should receive the authorization to access an object in certain periods, only if nobody else was ever authorized to access the same object in any instant within those periods.

Besides proposing a basic set of operators to specify temporal dependencies, we introduce a formalism to concisely express various types of dependencies. For example, a single statement can specify that a user can read all the files that another user can read, in a specific time interval.

A formal semantics is defined for temporal authorizations. The subject of Temporal Authorization Base (TAB) administration (by using *addRule* and *dropRule*) and the conflict situations are investigated and resolved.

The rest of this paper is as follows: Section 2 gives a brief introduction of the SBAC model and describes the model of time used throughout our work. In section 3, we represent our authorization rules in detail and offer the formal semantics of them. We also briefly describe the administration of the authorization base and conflict resolution in access decision point. Section 4 describes the related works on this topic and finally, in section 5, we conclude the paper.

2 Preliminaries

In this section we give a brief introduction of SBAC model, proposed by Javanmardi *et al*. [2], and introduce our model of time.

2.1 Introduction to SBAC

Fundamentally, SBAC consists of three basic components: Ontology Base, Authorization Base and Operations. Ontology Base is a set of ontologies: Subjects–Ontology (SO), Objects–Ontology (OO) and Actions–Ontology (AO).

By modeling the access control domains using ontologies, SBAC aims at considering semantic relationships in different levels of ontology to perform inferences to make decision about an access request. Authorization Base is a set of authorization rules in form of $(s,o,\pm a)$ in which *s* is an entity in SO, *o* is an entity defined in OO, and *a* is an action defined in AO. In other words, a rule determines whether a subject which presents a credential *s* can have the access right *a* on object *o* or not.

The main feature of the model is reduction of semantic relationships in ontologies to subsumption relation. Given two concepts *C* and *D* and a knowledge base Σ, $C \prec D$ denotes that *D* subsumes *C* in Σ. This reasoning based on subsumption proves that *D* (the subsumer) is more general than *C* (the subsumee).

By reducing all semantic relationships to the subsumption, the following propagation rules are enough.

- *Propagation in subjects domain*: given $(s_i, o, \pm a)$, if $s_j \prec s_i$ then $(s_j, o, \pm a)$.
- *Propagation in objects domain*: given $(s, o_i, \pm a)$, if $o_j \prec o_i$ then $(s, o_j, \pm a)$.
- *Propagation in actions domain*:
  - Given $(s, o, +a_i)$, if $a_j \prec a_i$ then $(s, o, +a_j)$.
  - Given $(s, o, -a_j)$, if $a_j \prec a_i$ then $(s, o, -a_i)$.

2.2 Model of Time

We assume a discrete model of time. In the database community a chronon usually identifies the smallest indivisible unit of time [3]. We can take a chronon as our time unit in this paper. On this basis, our model of time is isomorphic to the natural numbers □ with the total order relation ≤.

It is worthwhile to note that, we suppose that the response time of the access control system is trivial and thus we ignore the time duration required by the system to check whether a requested access is granted or denied. This assumption allows us to take an access request time as the access time recorded in the history.

3 Temporal Semantic Based Access Control Model

In this section we introduce our authorization model, Temporal Semantic base Access Control model (TSBAC), which is an extension of SBAC model. In our model, we extend the basic authorization model in two directions: adding authorization validation time interval, and associating a temporal expression over a *history base*.

3.1 Temporal Authorization Rules

In our model we consider a temporal constraint to be associated with each authorization. This constraint is based on the privileges granted to subjects of the system (on objects), or access requests denied, in a specific *time point* in the past. These elements of history are stored in *History Base*, in the form of $(t, s, o, +a)$ and $(t, s, o, -a)$. We refer to an authorization together with a temporal constraint as a temporal authorization rule. A temporal authorization rule is defined as follows.

Definition 1 (Temporal Authorization Rule): A temporal authorization rule is a triple $(\lfloor t_s, t_f \rfloor, (s, o, \pm a), F)$, where $t_s \in N$, $t_f \in N \cup \{\infty\}$ ($t_s \le t_f$) represents the authorization validation time interval, and formula *F* is a temporal constraint which is formally defined as in (1).

$$
\begin{aligned}
&F ::= true \mid false \mid done(s, o, a) \mid denied(s, o, a) \mid \\
&!F \mid F \wedge F \mid F \vee F \mid F \rightarrow F \mid F \leftrightarrow F \mid \\
&prev(F) \mid past\#(F) \mid H(F) \mid Fsb\#F \mid FabF \mid FssF \mid FduringF
\end{aligned}
\tag{1}
$$

Temporal authorization rule $\left(\left[t_s,t_f\right],(s,o,\pm a),F\right)$ states that subject *s* is allowed (or not allowed) to exercise access *a* on object *o* in the interval $\left[t_s,t_f\right]$, including time instants t_s and t_f, in the case that *F* is evaluated to true.

Definition 2 (Temporal Authorization Base (TAB)): A temporal authorization base (TAB) is a set of temporal authorization rules in the form of $\left(\left[t_1,t_2\right],(s,o,\pm a),F\right)$.

Definition 3 (History Base): A History Base is a set of authorizations and time points, in the form of $(t,s,o,+a)$ which means access *a* has been granted to subject *s* on object *o* at time point *t*, and $(t,s,o,-a)$ which means the system has denied access *a* on object *o* at time point *t* requested by subject *s*.

Definition 4 (Valid Authorization): an authorization $(s,o,\pm a)$ is valid at time *t* if one of the following situations occurred:

1. At time *t*, a temporal authorization rule $\left(\left[t_1,t_2\right],(s,o,\pm a),F\right)$ with $t_1 \le t \le t_2$ exists in TAB and *F* is evaluated to true based on the elements exist in *History Base* (in section 3.2 we define function *f* for performing such an evaluation),
2. There exists a temporal authorization rule $\left(\left[t_1,t_2\right],(s',o',\pm a'),F\right)$ in TAB with $t_1 \le t \le t_2$ in which *F* is evaluated to true, and $(s',o',\pm a')$ is derived from $(s,o,\pm a)$ following the inference rules of SBAC.

The intuitive meaning of temporal authorization rules is as follows. In these statements *auth* is representative of $(s,o,\pm a)$.

– $\left(\left[t_s,t_f\right],auth,true\right)$: Authorization *auth* is always valid in interval $\left[t_s,t_f\right]$.

– $\left(\left[t_s,t_f\right],auth,false\right)$: Authorization *auth* is always invalid.

– $\left(\left[t_s,t_f\right],auth,done(s,o,a)\right)$: Authorization *auth* is valid in all time instants *t*, in interval $\left[t_s,t_f\right]$ in which $done(s,o,a)$ is evaluated to true.

– $\left(\left[t_s,t_f\right],auth,denied(s,o,a)\right)$: Authorization *auth* is valid in all time instants *t*, in interval $\left[t_s,t_f\right]$ in which $denied(s,o,a)$ is evaluated to true.

– $\left(\left[t_s,t_f\right],auth,!F\right)$: Authorization *auth* is valid for each time instant *t* in interval $\left[t_s,t_f\right]$ in which *F* is *not* evaluated to true.

– $\left(\left[t_s,t_f\right],auth,F_1 \wedge F_2\right)$: Authorization *auth* is valid for each time instant *t* in the interval $\left[t_s,t_f\right]$ in which F_1 and F_2 are both evaluated to true.

– $\left(\left[t_s,t_f\right],auth,F_1 \vee F_2\right)$: Authorization *auth* is valid for each time instant *t* in the interval $\left[t_s,t_f\right]$ in which F_1 or F_2 or both of them are evaluated to true.

- $\left(\left[t_s, t_f\right], auth, F_1 \rightarrow F_2\right)$: Authorization *auth* is valid for each time instant *t* in the interval $\left[t_s, t_f\right]$ in which if F_1 is evaluated to true, then F_2 is also evaluated to true.
- $\left(\left[t_s, t_f\right], auth, F_1 \leftrightarrow F_2\right)$: Authorization *auth* is valid for each time instant *t* in the interval $\left[t_s, t_f\right]$ in which F_1 is evaluated to true if and only if F_2 is evaluated to true.
- $\left(\left[t_s, t_f\right], auth, prev(F)\right)$: Authorization *auth* is valid at the time of request (*t*) in interval $\left[t_s, t_f\right]$ if *F* is evaluated to true at the previous moment (*t-1*).
- $\left(\left[t_s, t_f\right], auth, past\#(F)\right)$: Authorization *auth* is valid at the time of request (*t*) in interval $\left[t_s, t_f\right]$ if *F* is evaluated to true, at least # times from t_s till *t*.
- $\left(\left[t_s, t_f\right], auth, H(F)\right)$: Authorization *auth* is valid at the time of request (*t*) in interval $\left[t_s, t_f\right]$ if *F* is evaluated to true in all time instants from t_s till *t*.
- $\left(\left[t_s, t_f\right], auth, F_1 sb\# F_2\right)$: Authorization *auth* is valid at the time of request (*t*) in interval $\left[t_s, t_f\right]$ if F_1 is evaluated to true, at least # times before the last occurrence of F_2, in interval $\left[t_s, t\right]$.
- $\left(\left[t_s, t_f\right], auth, F_1 abF_2\right)$: Authorization *auth* is valid at the time of request (*t*) in the interval $\left[t_s, t_f\right]$ if F_1 is evaluated to true in $t'\left(t_s < t' < t\right)$, then there exist a time point $t''\left(t' < t'' < t\right)$, in which F_2 is evaluated to true.
- $\left(\left[t_s, t_f\right], auth, F_1 ssF_2\right)$: Authorization *auth* is valid at the time of request (*t*) in interval $\left[t_s, t_f\right]$ if F_1 is evaluated to true in all the time points from the first occurrence of F_2 in interval $\left[t_s, t\right]$.
- $\left(\left[t_s, t_f\right], auth, F_1 duringF_2\right)$: Authorization *auth* is valid at the time of request (*t*) in interval $\left[t_s, t_f\right]$ if F_1 is not true before the first, or after the last time instant in which F_2 is true.

Another convention that could be useful here is the notion of parametric authorization rules. A parametric authorization rule is an authorization rule where keyword *all* appears for subjects, objects, or access rights in the authorizations. Keyword *all* is a parameter which denotes any subject, object, or access right depending on its position in the authorization.

3.2 Formal Semantics

To formalize the semantics of temporal authorization rules, we first define an evaluation function *f*. This function evaluates the predicate *F* of temporal authorization rules

at a time point *t* and based on the elements stored in *History Base*. Function *f* is defined as in (2).

F *is defined as in* (1);

we define an interpretation of function f *as follows:*

$$f\left(t, done\left(s,o,a\right)\right) = \begin{vmatrix} true & if\ \left(t,s,o,+a\right) \in HB \\ false & if\ \left(t,s,o,+a\right) \notin HB \end{vmatrix}$$

$$f\left(t, denied\left(s,o,a\right)\right) = \begin{vmatrix} true & if\ \left(t,s,o,-a\right) \in HB \\ false & if\ \left(t,s,o,-a\right) \notin HB \end{vmatrix}$$

$$f\left(t,true\right) = true; \qquad f\left(t,\ false\right) = \ false; \qquad f\left(t,!F\right) = \neg f\left(t,F\right);$$

$$f\ \left(t,F_1 \wedge F_2\right) = f\ \left(t,F_1\right) \wedge f\ \left(t,F_2\right); \qquad f\ \left(t,F_1 \vee F_2\right) = f\ \left(t,F_1\right) \vee f\ \left(t,F_2\right);$$

$$f\ \left(t,F_1 \to F_2\right) = f\ \left(t,F_1\right) \to f\ \left(t,F_2\right); \qquad f\ \left(t, prev\ \left(F\right)\right) = f\ \left(t-1,F\right);$$

$$f\left(t,F_1 \leftrightarrow F_2\right) = \left(f\left(t,F_1\right) \to f\left(t,F_2\right)\right) \wedge \left(f\left(t,F_2\right) \to f\left(t,F_1\right)\right);$$

$$f\ \left(t, past\ \#\left(F\right)\right) = \exists t_{i_1},...,t_{i_\#} \le t,\ \bigwedge_{k=1}^{\#}\left(f\ \left(t_{i_k},F\right)\right)$$

$$f\ \left(t,H\ \left(F\right)\right) = \forall t_i \le t,\ \ f\ \left(t_i,F\right)$$

$$f\ \left(t,F_1 sb\ \#F_2\right) = \exists t_2 \le t,\ \ \exists t_{i_1},...,t_{i_\#} \le t_2,\ \ f\ \left(t_2,F_2\right) \wedge \bigwedge_{k=1}^{\#} f\ \left(t_{i_k},F_1\right)$$

$$f\ \left(t,F_1 abF_2\right) = \left[\exists t_1 \le t, f\ \left(t_1,F_1\right)\right] \to \left[\exists t_2, t_1 \le t_2 \le t \wedge f\ \left(t_2,F_2\right)\right]$$

$$f\ \left(t,F_1 ssF_2\right) = \forall t_2 \le t, \left(f\ \left(t_2,F_2\right) \to \forall t_i, t_2 \le t_i \le t \to f\ \left(t_i,F_1\right)\right)$$

$$f\ \left(t,F_1 duringF_2\right) = \begin{bmatrix} \exists t_{\min}, t_{\min} \le t \wedge f\ \left(t_{\min},F_2\right) \wedge \left(\neg\exists t_x, t_x < t_{\min} \wedge f\ \left(t_x,F_2\right)\right) \wedge \\ \exists t_{\max}, t_{\max} \le t \wedge f\ \left(t_{\max},F_2\right) \wedge \left(\neg\exists t_y, t_{\max} < t_y \wedge f\ \left(t_y,F_2\right)\right) \end{bmatrix} \to \\ \left[\forall t_1, t_1 < t \to \left(f\ \left(t_1,F_1\right) \to t_{\min} \le t_1 \le t_{\max}\right)\right] \qquad (2)$$

Note that a temporal authorization rule can be removed from TAB and therefore not be applicable anymore. For formalizing this issue, we associate with each temporal authorization rule the time t_d at which it is removed. Note that time t_d is not a constant and it is not known from the former. We use it as shorthand for expressing the point, up to which, a temporal authorization rule is applicable.

By the definition of evaluation function *f* and by the assumption described above, the semantics of authorization rules are in (3). In the following, $grant\left(t,\left(s,o,a\right)\right)$ denotes subject *s* is granted to exercise action *a* on object *o* and analogously $deny\left(t,\left(s,o,a\right)\right)$ denotes the access request of *s* for exercising an access *a* on object *o* is denied.

$$\begin{aligned} &\left(\left[t_s,t_f\right],\left(s,o,+a\right),F\right) \Leftrightarrow \forall t\left(t_s \le t \le \min\left(t_f,t_d-1\right) \wedge\ f\ \left(t,F\right)\right) \to grant\left(t,\left(s,o,a\right)\right) \\ &\left(\left[t_s,t_f\right],\left(s,o,-a\right),F\right) \Leftrightarrow \forall t\left(t_s \le t \le \min\left(t_f,t_d-1\right) \wedge\ f\ \left(t,F\right)\right) \to deny\left(t,\left(s,o,a\right)\right) \end{aligned} \qquad (3)$$

3.3 Access Control and Conflict Resolution

The centric security mechanism in each system is an access control system. By receiving an access request in such a system, we need to make a decision whether to grant the requested access or deny it. Following the proposed model of temporal authorization in the previous sections, by receiving an access request (s_r, o_r, a_r) at time t, the access control system performs the following steps:

1. Determine the explicit and implicit valid authorization rules in TAB at time *t* (following the definition of valid authorization rules),
2. Extract the set of valid authorization rules like $\left(\left[t_s, t_f\right], (s, o, \pm a), F\right)$ which match the access request i.e. $s = s_r$, $o = o_r$, $a = a_r$ (we call this set, MVA),
3. If there exist just positive valid authorization rule(s) like $\left(\left[t_s, t_f\right], (s, o, +a), F\right)$ in MVA, grant the requested access,
4. If there exist just negative valid authorization rule(s) like $\left(\left[t_s, t_f\right], (s, o, -a), F\right)$ in MVA, deny the access request,
5. If there exist both positive and negative authorization rules in MVA, do conflict resolution (following the approach described in section 3.4) and follow the result,
6. If there is not any valid authorization rule, which matches the requested access, follow the default access policy,
7. Record $(t, s, o, +a)$ in case of the requested access is granted and $(t, s, o, -a)$ in case of the access request is denied.

In this model, the default access policy might be *positive* to grant all undetermined accesses or *negative* to deny them. The default access policy is determined by the administrator.

In access control, due to the modal conflicts between the valid matched authorization rules (in the set MVA); it is required to have a conflict resolution strategy to resolve the conflicts. The conflict might be a result of semantic relationships between the entities (i.e. subjects, objects, and actions) and applying the inference rules of SBAC model, or the sub-interval relation between authorizations (i.e. $\left[t_{s_2}, t_{f_2}\right]$ is a sub-interval of $\left[t_{s_1}, t_{f_1}\right]$).

The model supports two predefined strategies for conflict resolution; negative authorization rule takes precedence (NTP) strategy, and positive authorization rule takes precedence (PTP) strategy. Similar to default access policy, the conflict resolution strategy is determined by the administrator.

3.4 Temporal Authorization Base Administration

Authorization rules can be changed upon the execution of administrative operations. In this paper, we consider a centralized policy for administration of authorizations where administrative operations can be executed only by the administrator.

Administrative operations allow the administrator to add, remove, or modify (a *remove* operation followed by an *add* operation) temporal authorizations rules. Each temporal authorization rule in the TAB is identified by a unique label assigned by the system at the time of its insertion. The label allows the administrator to refer to a specific temporal authorization rule upon execution of administrative operations. A brief description of the administrative operations is as follows:

- *addRule*: To add a new temporal authorization rule. When a new rule is inserted, a label (*rule identifier* or *rid*) is assigned by the system.
- *dropRule*: To drop an existing temporal authorization rule. The operation requires as argument, the label of the rule to be removed.

4 Related Work

Access control systems for protecting Web resources along with credential based approaches for authenticating users have been studied in recent years [1]. With the advent of Semantic Web, new security challenges were imposed to security systems. Bonatti *et al.*, in[4] have discussed open issues in the area of policy for Semantic Web community such as important requirements for access control policies. Developing security annotations to describe security requirements and capabilities of web service providers and requesting agents have been addressed in [5]. A concept level access control model which considers some semantic relationships in the level of concepts in the object domain is proposed in [6]. The main work on SBAC, which is the basis for our model, is proposed in [7] by Javanmardi *et al.*. SBAC is based on OWL ontology language and considers the semantic relationships in the domains of subjects, objects, and actions to make decision about an access request.

The first security policy based on past history of events is introduced as Chinese Wall Security Policy (CWSP) [8]. The objective of CWSP is to prevent information flows which cause conflict of interest for individual consultants. Execution history also plays a role in Schneider's security automata [9] and in the Deeds system of Edjlali [10]. However, those works focus on collecting a selective history of sensitive access requests and use this information to constrain further access requests; for instance, network access may be explicitly forbidden after reading certain files. Another approach which considers the history of control transfers, rather than a history of sensitive requests is presented in [11].

In a basic authorization model, an authorization is modeled by a triple $(s, o, \pm a)$, interpreted as "subject *s* is authorized to exercise access right *a* on object *o*". Recently, several extensions to this basic authorization model have been suggested. One of them is the temporal extension of it which increases the expressive power of the basic authorization model [3, 12-15]. Bertino *et al*. [12] specification of temporal parameters. In the model proposed by Bertino *et al*. in [12], an authorization is specified as $(time, auth)$, where $time = [t_b, t_e]$ is a time interval, and $auth = (s, o, m, pn, g)$ is an authorization. Here, t_b and t_e represent the start and end times respectively, during which *auth* is valid. *s* represents the subject, *o* the object, and *m* the privilege. *pn* is a binary parameter indicating whether an authorization is negative or positive, and *g* represents the grantor of the authorization. This model also allows operations *WHENEVER*, *ASLONGAS*, *WHENEVERNOT*, and *UNLESS* on authorizations. For

example, *WHENEVER* can be used to express that a subject s_i can gain privilege on object *o* whenever another subject s_j has the same privilege on *o*. later Bertino *et al.* in [14] extended the temporal authorization model to support periodic authorization. They completed their research in [16] by presenting a powerful authorization mechanism that provides support for: (1) periodic authorizations (both positive and negative), that is, authorizations that hold only in specific periods of time; (2) user-defined deductive temporal rules, by which new authorizations can be derived from those explicitly specified; (3) a hierarchical organization of subjects and objects, supporting a more adequate representation of their semantics. From the authorizations explicitly specified, additional authorizations are automatically derived by the system based on the defined hierarchies.

5 Conclusions

In this paper, we presented TSBAC as an access control model that considers temporal aspects of access authorizations in semantic-aware environments like semantic web. The proposed model is an extension of SBAC model, which takes into account the semantic relationships in different levels of subjects, objects, and actions ontologies to perform inferences to make decision about an access request.

TSBAC adds two new elements to SBAC authorization model; temporal intervals of validity, and temporal expression over the history of accesses. This allows us to specify temporal dependencies between authorizations, in specific periods of time. The model is formally defined and its semantics is presented in this paper.

References

1. Samarati, P., Vimercati, S.C.: Access control: Policies, models, and mechanisms. In: Focardi, R., Gorrieri, R. (eds.) FOSAD 2000. LNCS, vol. 2171, pp. 137–196. Springer, Heidelberg (2001)
2. Javanmardi, S., Amini, A., Jalili, R.: An Access Control Model for Protecting Semantic Web Resources. In: Web Policy Workshop, Ahens, GA, USA (2006)
3. Bertino, E., Bettini, C., Samarati, P.: A temporal authorization model. In: Second ACM Conference on Computer and Communications Security, Fairfax, Va (1994)
4. Bonatti, P.A., Duma, C., Fuchs, N., Nejdl, W., Olmedilla, D., Peer, J., Shahmehri, N.: Semantic web policies: a discussion of requirements and research issues. In: Sure, Y., Domingue, J. (eds.) ESWC 2006. LNCS, vol. 4011, pp. 712–724. Springer, Heidelberg (2006)
5. Rabitti, F., Bertino, E., Kim, W., Woelk, D.: A Model of Authorization for Next-Generation Database Systems. ACM TODS 16 (1991)
6. Qin, L., Atluri, V.: Concept-level access control for the Semantic Web. In: 2003 ACM workshop on XML security (2003)
7. Javanmardi, S., Amini, A., Jalili, R., Ganhisafar, Y.: SBAC: A Semantic-Based Access Control Model. In: NORDSEC 2006 (2006)
8. Brewer, D.F.C., Nash, M.J.: The Chinese Wall Security Policy. In: IEEE Symposium on Security and Privacy, Oakland, California (1989)

9. Dias, P., Ribeiro, C., Ferreira, P.: Enforcing History-Based Security Policies in Mobile Agent Systems (2003)
10. Edjlali, G., Acharya, A., Chaudhary, V.: History-based access control for mobile code. In: 5th ACM conference on Computer and communications security (1998)
11. Abadi, M., Fournet, C.: Access control based on execution history. In: 10th Annual Network and Distributed System Security Symposium (2003)
12. Bertino, E., Bettini, C., Ferrari, E., Samarati, P.: A temporal access control mechanism for Database Systems. IEEE Trans. Knowl. Data Eng. 8, 67–80 (1996)
13. Thomas, R.K., Sandhu, R.S.: Sixteenth National Computer Security Conference. Baltimore, Md. (1993)
14. Bertino, E., Bettini, C., Ferrari, E., Samarati, P.: An access control model supporting periodicity constraints and temporal reasoning. ACM Trans. Database Systems 23, 231–285 (1998)
15. Ruan, C.: Decentralized Temporal Authorization Administration. CIT/27/2003 (2003)
16. Bertino, E., Bonatti, P.A., Ferrari, E., Sapino, M.L.: Temporal authorization bases: From specification to integration. Journal of Computer Security 8, 309–353 (2000)

A Review on Concepts, Algorithms and Recognition-Based Applications of Artificial Immune System

Shahram Golzari[1,2], Shyamala Doraisamy[1], Md Nasir B. Sulaiman[1], and Nur Izura Udzir[1]

[1] Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, 43400, Serdang, Selangor, Malaysia
[2] Electrical and Computer Engineering Department, Hormozgan University, Bandarabbas, Iran
golzarihormozi@yahoo.com,
{shyamala,nasir,izura}@fsktm.upm.edu.my

Abstract. This paper reviews the concepts and some basic algorithms of artificial immune system as a bio inspired computational model and considers works that have been done based on the learning and recognition capabilities of artificial immune system.

Keywords: Immune System, Artificial Immune System, Recognition, Learning.

1 Introduction

This paper presents review on the artificial immune system (AIS). Artificial immune system is a computational method inspired by the biology immune system. It is progressing slowly and steadily as a new branch of computational intelligence and soft computing [1-3]. From information processing view point, the biology immune system is a parallel and distributed adaptive system with decentralized control mechanism. It uses signaling, learning and memory to solving recognition and classification tasks. It learns to recognize relevant patterns and remember patterns that have been seen previously. These remarkable information processing abilities of the immune system provide several important aspects in the field of computation.

Therefore, recently a lot of researchers are interested in the artificial immune system. Nowadays it is covers several application areas such as machine learning, pattern recognition and classification, computer virus detection, anomaly detection, optimization, robotics and etc. This paper reviews concepts, some algorithms and application of this field. Paper is organized as follows: section 2 briefly reviews the concept of biological immune system. Section 3 describes the fundamentals of artificial immune systems and discusses about some basic artificial immune algorithms. Section 4 introduces and considers some works that use artificial immune system for solving problems based on recognition capability of artificial immune system. The paper is concluded in section 5.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 569–576, 2008.

2 Biological Immune System [2,3]

Immune system defenses our body against foreign attack. This defense process is interesting due to its several computational capabilities, as will be discussed throughout this section.

The immune system is composed of many molecules, cells, and organs. The components of immune system are distributed in the all of our body. There is no central organ controlling the functioning of the immune system. The main task of the immune system is to protect our organism against foreign attacker elements such as viruses and bacteria that cause to diseases and also against non normal itself cells such as cancer cells. Immunologists name each element that can be recognized by the immune system as antigen (Ag). They divide antigens to two major categories: self and nonself antigens. All cells that originally belong to our body and are harmless to its functioning are named as self antigens and others named as nonself antigens. The immune system can distinguish between self and nonself via the process called self/nonself discrimination and performed based on pattern recognition tasks.

The immune system can recognize antigens by using receptor molecules that exist on surface of immune cells. There are two groups of immune cells: T-cells and B-cells. B-cell receptor is named as BCR or antibody (Ab) and T-cell is named as TCR. The differences between T-cells and B-cells are in the used mechanisms to recognize antigens and also in their tasks. B-cells can recognize all antigens in the blood stream but T-cells require antigens to be presented by other accessory cells.

Immune system needs to do antigenic recognition for activation and generates an immune response. The process follows some steps. At first, the receptor molecule recognizes an antigen with a certain affinity. In fact, reorganization is the binding between receptor and antigen. If the affinity is greater than a given threshold, named affinity threshold, then the immune system is activated and goes to generate immune responses.

Three theories in immunology are attracted by computer scientist: negative selection, clonal selection and immune network theory. We describe each of them briefly in the next.

Negative selection occurs in the thymus. Thymus is an organ in behind of the breastbone. T-cells migrate to the thymus after their generation. The purpose of this immigration is maturation. During the maturation process some T-cells can recognize self antigens. These T-cells are removed from the T-cells population and others move to lymphatic system.

Clonal selection theory explains about the function of immune system after nonself antigenic recognition by a B-cell receptor (antibody) with a certain affinity. This antibody is selected to proliferate and produce antibodies in high rate (cloning). In proliferation process, cells divide themselves. The proliferation rate of a cell has direct relation to its affinity with antigen. The more affinity cause to more clones. During reproduction, the hypermutation process is also done on antibodies. Mutation rate has inverse relation with affinity between antibody and antigen: the higher affinity, the lower the mutation rate. The mutation is done to increase the affinity and also selective pressure of antibodies. After finishing these cloning and hypermutation process, the immune system has the antibodies with improving affinity. Finally, some antibodies with higher affinities are selected to remain in the system for long time.

These antibodies are named memory cells. If the system is attacked by the same type or similar antigen in the future, the memory cells are activated to present more efficient response. The combination of mutation and selection is named as affinity maturation; response generated by immune system named secondary response.

In the immune network theory, the elements of immune system can react together even in the absence of external stimulation. Here, the immune system is a network of B-cells and the immune cells can recognize and suppress each other to network stay in stable situation. The immune cells also can recognize nonself antigen. Several immunologists have refused this theory. But, it has proved itself as powerful model for computational systems because of its relevant computational issues.

3 Artificial Immune System

This section introduces the Artificial Immune System (AIS) and their characteristic as a computational intelligence model. There are different definitions for AIRS. Dasgupta [1] defines AIS as: "Artificial immune system is intelligent methodology inspired by the immune system toward real-world problem solving". While based on the deCastro and Timmis [3]: "AIS are adaptive systems inspired by theoretical immunology and observed immune functions, principles and models, which are applied to complex problem domains". This can be concluded from the definitions that AIS uses the concepts and processes of immune system to solve real and complex problem. Immune system is appealed for computer scientist for its computational capabilities.

From natural computation view AIS is a bio-inspired computational model. This model should be has a framework to design the computational algorithms. deCastro and Timmis[3] have proposed the framework for bio-inspired computational models. This requires at least the following basic elements:

> "-A representation for the components of the system.
> -A set of mechanism to evaluate the interaction of individuals with the environment and each other. The environment is usually simulated by a set of input stimuli, one or more fitness functions.
> -Procedures of adaptation that govern the dynamics of the system. How its behavior varies over time [3]."

This structure has known as layered structure [3]. Based on this structure to design AIS, we need to application domain or target function at first. When the domain is clarified, the mechanism is chosen to represents the components of the system. The components of system are antibodies and antigens. After that, affinity measures are used to implement the interactions between element the system. The final layer involves the use of algorithms with govern the behavior of the system, such as negative selection, clonal selection and immune network algorithms [3-5]. In the following sections the common representation methods, affinity measures and algorithms, which have been used frequently in the AIS literature, will be introduced.

Stepney et al. [5] have proposed a conceptual framework for bio-inspired computational domain and adopted it for AIS. The proposed framework includes various components to cover the interdisciplinary researches.

3.1 Antibody/ Antigen Representation

The first step in layer framework is chosen the representation mechanism for antibodies and antigens. In general form antibodies and antigens are represented by an L-dimensional vector. L is the length of the vectors that is equal to the number of attributes of antigen or antibody. Notations <Ab> and <Ag> are used for Antibody and antigen vectors respectively. Tree types of representation are used by researchers to represent the antibodies and antigens mostly: binary representation, numeric representation and categorical representation. As we mentioned, representation type is chosen based on the application domain.

In the binary representation, antibody and antigen are modeled by the set of bits and the components interact to each others by using bitwise operators. In the numeric representation, all the attributes of the antibody and antigen vectors are integer or real valued. Numeric distance measures are used to calculate the matching between an antibody and an antigen. Euclidean distance and Manhattan distance have used in several researches as affinity measures for numeric representation. All coordinates of antibody and antigen vectors are categorical or nominal values in the categorical representation.

In many applications, datasets have the data attributes with the different data types. Above representations are not suitable for these datasets; because they can represent only one type of data. Hybrid representations should be used for these applications; but hybrid representation has not used in AIS community very well and researchers use the artificial method to map different data types to one representation strategy [6].

3.2 Affinity Functions

The affinity function selection has closed relation to the representation mechanism. For the binary representation, the best affinity function is the well known Hamming distance or its complement. Hamming distance counts the number of bit positions that have same values in antigen and antibody. Other affinity functions have also been used, in particular the r-contiguous bits rule. The majority of the AIS literature uses the Euclidean distance for numeric data. Others distance measurements such as the Manhattan distance is also used in some researches. In categorical representation, the antigen and antibody have the same value or different value. If they have same values, the affinity function gets the value that shows similarity. Otherwise, the affinity function gets the value that shows the difference. These values are selected based on application domain and data.

3.3 Procedures and Algorithms

In this section, basic algorithms of AIS are described. Based on the immune processes many computational algorithms have been generated and used that some of them use combination of different processes of immune system. Basic algorithms can be modified to use in new application domain. In literature [1-3], AIS based algorithms are divided to major category: population based and network based. Network based algorithms use the concepts of immune network theory; while population based algorithms use the other theories such as clonal selection and negative selection.

Negative selection algorithm. The purpose of pattern recognition is to find the useful relations and patterns from the information. In common approaches, theses patterns recognized from the stored information about them. Negative selection proposes alternative approach to perform pattern recognition. In this approach, the complement set of information about the patterns are stored and patterns recognized regarding these information. The focus of negative selection algorithm is on anomaly detection problems such as computer and network intrusion detection. Beside it is used in time series prediction, image inspection and segmentation, and hardware fault tolerance [7].

Given an appropriate problem representation, define the set of patterns to be protected and call it the self-set (**P**). Based upon the negative selection algorithm, generate a set of detectors (**M**) that will be responsible to identify all elements that do not belong to the self-set, i.e., the nonself elements. The negative selection algorithm runs as follows [8]:

1. Generate random candidate elements (**C**) using the same representation adopted;
2. Compare (match) the elements in **C** with the elements in **P**. If an element of **P** is recognized by an element of **C**, then discard this element of **C**; else store this element of **C** in the detector set **M**.

After generating the set of detectors (**M**), the next stage of the algorithm consists in monitoring the system for the presence of nonself patterns. In this case, assume a set **P*** of patterns to be protected. This set might be composed of the set **P** plus other new patterns, or it can be a completely novel set. For all elements of the detector set, that corresponds to the nonself patterns, check if it recognizes (matches) an element of **P*** and, if yes, then a nonself pattern was recognized and an action has to be taken.

Clonal selection algorithm. There are different views about the clonal selection in the literature. Some researches [7] believe that clonal selection is the genetic algorithms with a crossover operator; while others [9] believe that genetic algorithm can not cover all aspects of clonal selection such as affinity proportional reproduction and mutation and hypermutation. Latter group proposed a clonal selection algorithm, named CLONALG that uses the basic processes involved in clonal selection [9]. This algorithm was initially proposed to perform pattern recognition and then adapted to solve multi-modal optimization tasks. Given a set of patterns to be recognized (**P**), the basic steps of the CLONALG algorithm are as follows [10]:

1. Randomly initialize a population of individuals (**M**);
2. For each pattern of **P**, present it to the population **M** and determine its affinity (match) with each element of the population **M**;
3. Select $n1$ of the best highest affinity elements of **M** and generate copies of these individuals proportionally to their affinity with the antigen. The higher the affinity, the higher the number of copies, and vice-versa;
4. Mutate all these copies with a rate proportional to their affinity with the input pattern: the higher the affinity, the smaller the mutation rate, and vice-versa.

5. Add these mutated individuals to the population **M** and re-select *n*2 of these maturated (optimized) individuals to be kept as memories of the system;
6. Repeat Steps 2 to 5 until a certain criterion is met, such as a minimum pattern recognition or classification error.

4 Recognition Based Applications

There are several works that have used artificial immune system and its recognition capability. The use of the negative and clonal selection algorithms have been widely tested on this application. The former because it is an inherent anomaly detection system, constituting a particular case of pattern recognition device. The latter, the clonal selection algorithm, has been used in conjunction to negative selection due to its learning capabilities. Negative selection is used for problem of anomaly detection, such as computer security, network intrusion detection, hardware fault tolerance and image inspection. Clonal selection algorithm is used in areas such as machine learning, optimization, classification, data mining and clustering. Artificial network is used in fault detection, learning systems, multi agent systems and robotics.

Based on the immune network theory Kayama et al. [12] generated a diagnosis method for sensors in control systems that consist of training model and diagnosis model. Dasgupta and Forrest [13] used the negative selection algorithm to detect faults in temperature sensors. Dasgupta et al. [14] presented multi level immune learning algorithm (MILA) that combines several immunological features and he applied this algorithm for anomaly detection and pattern recognition. Stibor et al. [15] used self-nonself discrimination concept and real value vector representation for anomaly detection.

Forrest et al. [8] used negative selection algorithm to protect computer system against the worms and viruses. Several works have been published [7,8] pursuing the problem of developing an artificial immune system that is distributed, robust dynamic , divers and adaptive, with application to computer network security.

Carter [16] presents the immune system as a model for recognition and classification. Cao and Dasgupta [17] used the AIS for recognizing the chemical spectrums. Castro and Zuban [10] propose CLONALG algorithm using a selection of survivors proportional to fitness, in which only the most apt anti bodies survive the next generation. This algorithm has several interesting features such as population size dynamically adjustable, exploitation and exploration of the search space, defined stopping criterion, capability of maintaining local optima solution and location of multiple optima [10-11]. They used this algorithm for pattern recognition and optimization. Temmis et al. [18] presented an another machine learning based on AIS called AINE that uses artificial recognition ball to represent a number of similar B-cells. Alexandro and Filho [19] investigated a new artificial system and apply it to pattern recognition. Their model was based on clustering of antibodies and generational antibody population. The above works are only some works in the artificial immune system field that explore the capability of it as a computational model.

5 Conclusions

In this paper artificial immune system has been reviewed as a bio-inspired computational and soft computing model. The basic concepts of immune system have been introduced and three classes of artificial immune system algorithms to perform pattern recognition: negative selection, clonal selection and immune network models, have been reviewed. In negative selection, a pattern recognition system is designed by learning information about the complement set of the patterns to be recognized. Clonal selection algorithms learn to recognize patterns through an evolutionary-like procedure. Finally, immune network models are peculiar because they carry information about the patterns to be recognized and, also, they have knowledge of themselves. Then, some works that have used artificial immune system have been presented.

References

1. Dasgupta, D.: Advances in Artificial Immune Systems. IEEE Computational Intelligence Magazine, 40–49 (November 2006)
2. de Castro, L.N., Timmis, J.: Artificial Immune Systems as a novel Soft Computing Paradigm. Soft. Comp. J. 7, 7 (2003)
3. de Castro, L.N., Timmis, J.: Artificial Immune Systems: A New Computational Intelligence Approach. Springer, Heidelberg (2002)
4. Andrews, P.S., Timmis, J.: Inspiration for the next generation of artificial immune systems. In: Proceeding of Fourth International Conference on Artificial Immune Systems, pp. 126–138 (2005)
5. Stepney, S., Smith, R.E., Timmis, J.: Towards a conceptual framework for artificial immune systems. In: Proceedings of Third International Conference on Artificial Immune Systems, pp. 53–64 (2004)
6. Freitas, A.A., Timmis, J.: Revisiting the Foundations of Artificial Immune Systems: A Problem-oriented Perspective. In: Proceeding of Second International Conference on Artificial Immune Systems (2003)
7. Hofmeyr, S.A., Forrest, S.: Architecture for an artificial immune system. Evolutionary Computation 8(4), 443–473 (2000)
8. Forrest, S., Perelson, A.S., Allen, L., Cherukuri, R.: Self-Nonself Discrimination in a Computer. In: Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy. IEEE Computer Society Press, Los Alamitos (1994)
9. de Castro, L.N., Von Zuben, F.J.: The clonal selection algorithm with engineering applications. In: Proceedings of GECCO 2000, pp. 36–39 (2000)
10. de Castro, L.N., Von Zuben, F.J.: Learning and Optimization Using the Clonal Selection Principle. IEEE Transactions on Evolutionary Computation 6(3), 239–251 (2002)
11. de Castro, L.N., Von Zuben, F.J.: aiNet: An artificial immune network for data analysis. In: Abbass, H.A., Sarker, R.A., Newton, C.S. (eds.) Data Mining: A Heuristic Approach, pp. 231–259. Idea Group Publishing, USA (2001)
12. Kayama, M., Sugita, Y., et al.: Distributed Diagnosis System Combining the Immune Network and Learning Vector Quantization. In: Proceeding of 21st International Conference on Industrial Electronics, Control, and Instrumentation, Orlando, FL, pp. 1531–1536 (1995)

13. Dasgupta, D., Forrest, S.: Tool Breakage Detection in Milling Operations Using A Negative-Selection Algorithm. Technical Report CS95-5, Department of Computer Science, University of New Mexico (1995)
14. Dasgupta, D., Yu, S., Majumdar, N.S.: MILA—multilevel immune learning algorithm. In: Cantú-Paz, E., Foster, J.A., Deb, K., Davis, L., Roy, R., O'Reilly, U.-M., Beyer, H.-G., Kendall, G., Wilson, S.W., Harman, M., Wegener, J., Dasgupta, D., Potter, M.A., Schultz, A., Dowsland, K.A., Jonoska, N., Miller, J., Standish, R.K. (eds.) GECCO 2003. LNCS, vol. 2723, pp. 183–194. Springer, Heidelberg (2003)
15. Stibor, T., Timmis, J., Eckert, C.: A comparative study of real-valued negative selection to statistical anomaly detection techniques. In: Proceeding of Fourth International Conference on Artificial Immune Systems, pp. 262–275 (2005)
16. Carter, J.H.: The Immune System as a Model for Pattern Recognition and Classification. Journal of the American Medical Informatics Association 7(1), 28–41 (2000)
17. Cao, Y., Dasgupta, D.: An Immunogenetic Approach in Chemical Spectrum Recognition. In: Ghosh, T. (ed.) Advances in Evolutionary Computing. Springer, Heidelberg (2003)
18. Timmis, J., Neal, M., Knight, T.: AINE: Machine Learning Inspired by the Immune System. IEEE Transactions on Evolutionary Computation (2002)
19. Alexandrino, J.L., Filho, B.C.: Investigation of a New Artificial Immune System model applied to pattern Recognition. In: Proceeding of the 6th IEEE International Conference on Hybrid Intelligent Systems (2006)
20. Timmis, J.: Artificial Immune Systems: a novel data analysis techniques inspired by the immune network theory. PhD Thesis, University of Wales, Aberystwyth (2001)

Incremental Hybrid Intrusion Detection Using Ensemble of Weak Classifiers

Amin Rasoulifard, Abbas Ghaemi Bafghi, and Mohsen Kahani

Department of Computer Science and Engineering, Faculty of Engineering
Ferdowsi University of Mashhad, Mashhad, Iran
am_ra84@stu-mail.um.ac.ir, {ghaemib,kahani}@um.ac.ir

Abstract. In this paper, an incremental hybrid intrusion detection system is introduced. This system combines incremental misuse detection and incremental anomaly detection. It can learn new classes of intrusions that do not exist in the training dataset for incremental misuse detection. As the framework has low computational complexity, it is suitable for real-time or on-line learning. Also experimental evaluations on KDD Cup dataset are presented.

Keywords: Hybrid intrusion detection system, Ensemble of Weak Classifiers, Incremental learning, KDD Cup 99 Dataset.

1 Introduction

Misuse detection systems use patterns of well-known attacks or weak spots of the system to identify intrusions. The main shortcoming of such systems[1-3] are the necessity of hand-coding of known intrusion patterns and their inability to detect any future(unknown) intrusions not matched with the patterns stored in the system. Anomaly detection systems, on the other hand, firstly establish normal user behavior patterns (profiles) and then try to determine whether deviations from the established normal profiles can be flagged as intrusions. The main advantage of anomaly detection systems is that they can detect new types of unknown intrusions [4-6].

Weak classifiers are those that obtain 50 percent classification accuracy on its own training data [7]. *Ensembles* are combinations of several models whose individual predictions are combined in some manner (e.g., averaging or voting) to form a final prediction [8].

Several hybrid intrusion detection systems have been proposed for combining misuse detection and anomaly detection [9-17]. We propose a hybrid intrusion detection system which combines the incremental misuse intrusion detection and incremental anomaly detection. In addition, when the intrusion detection dataset is so large that whole dataset can't be loaded into the main memory, the original dataset can be partitioned into several subsets, and then the detection model is dynamically modified according to other training subsets after the detection model was built on first subset.

The rest of the paper is organized as follows: related works is presented in section 2, the proposed incremental IDS is presented in section 3, KDD Cup 99 Dataset is presented in Section 4, experimental evaluation is presented in section 5, comparison

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 577–584, 2008.

to other algorithms is presented in section 6, conclusion is presented in section 7 and finally we conclude the paper in the references section.

2 Related Works

ADAM (Audit data analysis and mining) is a hybrid on-line intrusion detection system which uses association rules for detecting intrusions [9]. The Next Generate Intrusion Expert System (NIDES) is a hybrid system [10]. It is consists of rule-based misuse detection and anomaly detection that use statistical approaches.

The random forest algorithm used for hybrid intrusion detection system is proposed in [11]. It uses ensemble of classification tree for misuse detection and use proximities to find anomaly intrusions. FLIPS is a framework which uses hybrid approach for intrusion prevention systems [12]. The core of this framework is an anomaly-based classifier that incorporates feedback from environment to both tune its model and automatically generate signatures of known malicious activities.

In [13], the hierarchical layering distributed intrusion detection system and response is proposed. In this paper, the authors propose to use analysis tool (anomaly detection) at first to generate malicious activities, and then apply misuse detection on these activities for integrating to the network computing environments. In [14], a serial combination of misuse detection and anomaly detection is proposed. They propose a set model to formally describe anomaly and misuse intrusion detection results. In [15], the authors propose a novel intrusion detection system architecture utilizing both anomaly and misuse detection approaches. This hybrid intrusion detection system consists of an anomaly detection module, a misuse detection module and a decision support system combining the results of these two detection models.

In [16], the authors proposed a hybrid intrusion detection system which combines the advantages of low false-positive rate of signature-based intrusion detection system and the ability of anomaly detection system to detect novel attacks. In [17], the authors propose a multi-level hybrid intrusion detection system that uses a combination of tree classifiers and clustering algorithms to detect intrusions.

In the proposed incremental hybrid intrusion detection system, we use ensemble of weak classifiers for the incremental misuse detection component and use the on-Line K-Mean algorithm for the incremental anomaly detection component.

3 The Proposed Incremental IDS

The proposed framework consists of two phases: on-line phase and off-line phase. Misuse intrusion detection component is used in the on-line phase. We use ensemble of weak classifiers to implement misuse detection component [18]. It has the ability to learn new classes of intrusions that does not exist in the previous data which used for training the existing classifiers. In other words, the new classes of intrusions can be learned in a supervised mode. It is also suitable for learning known intrusions in on-line mode because of using an ensemble of weak classifiers with lower computational complexity.

In the off-line section of our framework, we use on-line k-mean algorithm to implement anomaly detection [20]. It can identify new unknown intrusions and can learn them incrementally. The new intrusions identified by anomaly detection component must be applied to the misuse intrusion detection component in the next learning phase. Therefore, we must determine the class types of new intrusions. For this reason, another component must be used. Any supervised or unsupervised clustering algorithms can be used for this component. In our experiment, we manually determine the class types of intrusions detected as attack by anomaly detection component. In the future work another component will be used to determine the class types of new unknown intrusions automatically. Fig. 1 shows the proposed framework which has the following components:

WL: Our framework requires a base classifier to generate a group of **weak Learner** (weak hypothesis) designed before hand. Weak Learner can obtain a 50% correct classification performance on its own training dataset. We use multi layer perceptron as a base classifier for generating weak hypotheses.

$\sum$ **: Weighted Majority voting:** used for calculating the final classification accuracy based on the classification accuracy of the weak hypotheses [8].

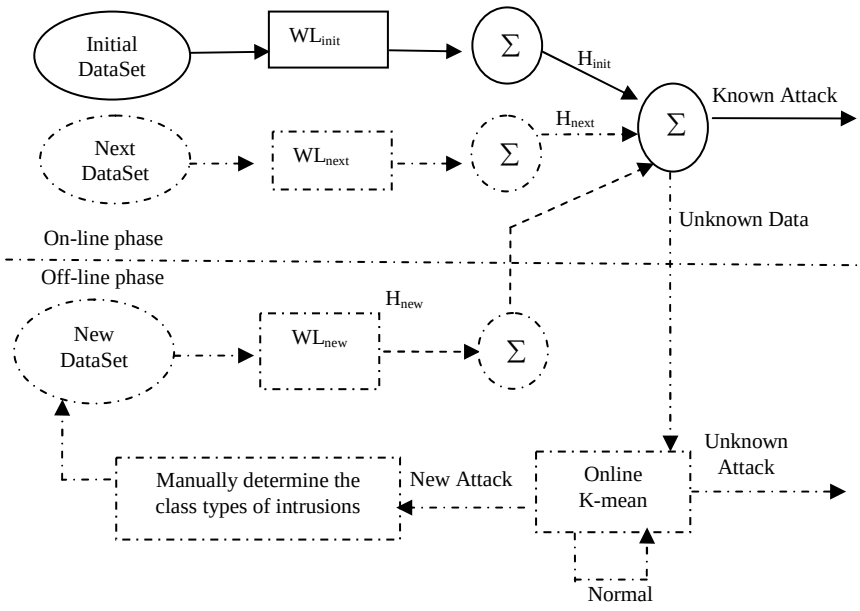


Fig. 1. Incremental hybrid intrusion detection system

4 KDD Cup 99 Dataset

To simulate the presented ideas, we used the 1999 DARPA Intrusion Detection Evaluation program data provided by MIT Lincoln Labs. The TCP dump raw data

was processed into connection records, which are about five million connection records. The data set contains 24 attack types. The attacks fall into four main categories as follows:

Denial of Service (DOS): Attacker makes some computing or memory resources too busy or too full to handle legitimate requests, or denies legitimate users access to a machine.

Remote to User (R2L): Attacker who does not have an account on a remote machine sends packets to that machine over a network and exploits some vulnerability to gain local access as a user of that machine.

User to Root (U2R): Attacker starts out with access to a normal user account on the system and is able to exploit vulnerability to gain root access to the system.

Probing: Attacker scans a network of computers to gather information or find known vulnerabilities. An attacker with a map of machines and services that available on a network can use this information to look for exploits.

5 Experimental Evaluations

For simulations, the weak learner used to generate individual hypotheses was a single hidden layer MLP with a 41 neuron in input layer, 41 hidden layer nodes and 4 nodes in output layer. The 4 nodes in output layer correspond to the four class type of intrusions. The mean square error goals of all MLPs were preset to values of 0.02 to prevent overfitting and to ensure sufficiently weak learning. To construct the profile of normal activities, we set the number of clusters in On-Line K-Mean algorithm equal to 70.

For validating the effectiveness of incremental hybrid intrusion detection using ensemble of weak classifiers, the following experiments are done. The original training sample obtained from the KDD Cup 99 dataset was used in our study as training set and entire labeled test set is used for testing set. We use intrusions of training set to generate the model of incremental misuse component and use normal activities of training set to construct the profile of normal activities.

In order to generate the initial model of our framework, the following scenario is done. In order to make the anomaly detection component, we elicit the normal instances from training dataset for constructing the profile of normal activities. The remaining intrusions instances used as intrusion dataset to make the misuse detection component. We select 10% of intrusion dataset consisting of about 400000 instances which contains four class types of intrusions as initial dataset. This dataset used to make an initial model of misuse detection component.

After getting the initial model of our framework, we apply the 90% remaining attacks dataset to add additional model to the initial model. We use a threshold equal to 100000 instances for constructing the next dataset. It will be used to construct the next model or classifier (H). The ensemble of existing classifiers is used for misuse detection component in the next iterations.

After we get the new model of misuse detection component based on the training dataset, we test the model on testing dataset for evaluating the effectiveness of our incremental intrusion detection system. The results of simulation show that when new unknown data become available (new classifier is generated), the classification performance of ensemble approach on testing dataset increased. In other words, the framework can learn new information in the next iterations when becomes available in current iteration.

Fig. 2 shows that adding the additional training sample to generate classifier (H) caused an improvement in the detection rate of misuse detection on test dataset from 45.3 to 87.8. This demonstrates its incremental learning capability even when instances of new classes are introduced in subsequent training data. So, our hybrid intrusion detection system when introduced can learn new information incrementally.

As indicated in Fig. 3, there are instances of data in remaining dataset which detected as intrusions by anomaly detection. These instances were not predicted by misuse detection component. This means that combining misuse detection and anomaly detection can detect more intrusions than each of them individually. These intrusions will be learned in the next iterations. It is the main interest of ensemble learning using weak classifiers for incremental learning and demonstrates the adaptively and efficiency of presented incremental hybrid intrusion detection system.

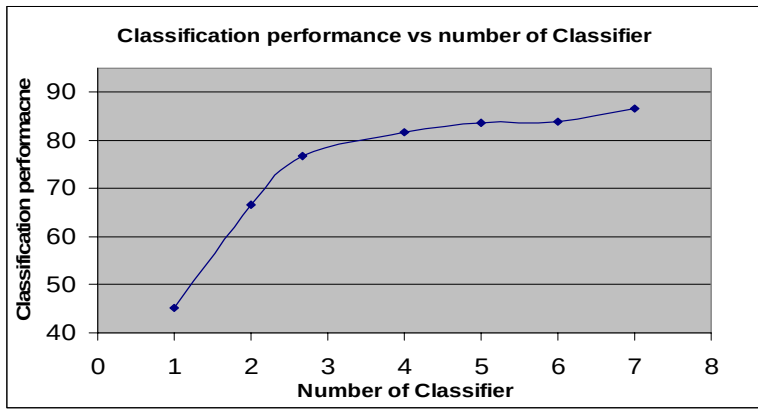


Fig. 2. Classification performance vs number of classifier

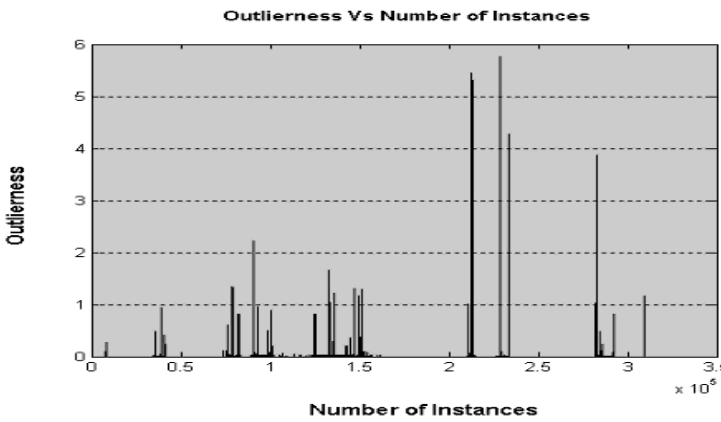


Fig. 3. Outlierness Vs Number of Instances

6 Comparison to Other Algorithm

6.1 Detection Rate Comparison

As indicated in Table1, we compare the performance classification of our framework with IEIDSLA [19] framework which running on KDD Cup dataset 99 because the contribution of both papers are based on incremental learning. We improve the detection rate on test dataset from 45.3% to 87.8% percent while the IEIDSLA improve the performance classification from 77.5% to 84.6%. It can be easily seen that our framework has higher detection rate while it used the weak classifiers with lower detection rate. This means that our framework can start with a small value of data that is available then gradually learn new information when become available.

Table 1. Comparison of Different Algorithms

Framework	Detection Rate (From)	Detection Rate (To)
IEIDSLA[19]	77.5%	84.6%
Our framework	45.3%	87.8%

6.2 Computational Complexity Comparison

After analyzing the LEARN++ algorithm[18], we calculate that in training phase of our framework, the computational complexity of initial model in the on-line phase is $O(nT_k\alpha)$, where n is the number of instances, T_k is the number of weak hypotheses that must be generated, and α is the complexity of BaseClassifier which in our framework is simple multi layer perceptron. For testing phase, computational complexity of our framework is $O(n\alpha')$, where n is the number of test instances, and α' is the complexity of BaseClassifier in testing phase. As to ANN, the computational complexity of the training phase depends on the distribution of dataset, and in the worst case it is $O(n^2M^2)$, which is higher than Learn++. M is the number of decision stumps. In a word, Learn++ generally possesses lower computational complexity than strong ANN, especially in training phase.

Clustering algorithms can be divided into two categories [20]: similarity based and centroid based. Similarity algorithms have a complexity at least $O(N^2)$, where N being the number of instances. In contrast centroid-based algorithms have a complexity of $O(NKM)$, where K is the number of clusters, M is the number of batch iteration and N is the number of instances. The on-line k-mean algorithm is a centroid-based which can be a desirable choice for on-line learning. Because it has high clustering quality, relatively lower complexity and fast convergence.

7 Conclusion

In the case that the intrusion detection instances are updated continually and infinitely, the static model learned on the initial training dataset unable to update the

profile of model dynamically. To make the intrusion detection model more adaptive to the network behavior, we present an incremental hybrid intrusion detection model based on the ensemble of weak classifiers. The detecting model can incorporate new instances continually, and therefore enhance generalization performance of detecting model. The research of this paper will have an important significance for building an efficient and applicable intrusion detection system.

References

1. Mounji, A., Charlier, B.L., Zampuniéris, D., Habra, N.: Distributed audit trail analysis. In: Balenson, D., Shirey, R. (eds.) Proceedings of the ISOC 1995 symposium on network and distributed system security, pp. 102–112. IEEE Computer Society, Los Alamitos (1995)
2. Lindqvist, U., Porras, P.A.: Detecting computer and network misuse through the production-based expert system toolset (PBEST). In: Gong, L., Reiter, M. (eds.) Proceedings of the 1999 IEEE symposium on security and privacy, pp. 146–161. IEEE Computer Socitey, Los Alamitos (1999)
3. Ilgun, K., Kemmerer, R.A., Porras, P.A.: State transition analysis: A rule-based intrusion detection approach. IEEE Transactions on Software Engineering 21(3), 181–199 (1995)
4. Neri, F.: Comparing local search with respect to genetic evolution to detect intrusions in computer networks. In: Proceedings of the 2000 Congress on Evolutionary Computation, Mar-seille, France, July 2000, vol. 1, pp. 238–243. IEEE, Los Alamitos (2000)
5. Guan, J., Liu, D.X., Cui, B.G.: An induction learning approach for building intrusion detection models using genetic algorithms. In: Proceedings of Fifth World Congress on Intelligent Control and Automation WCICA, vol. 5, pp. 4339–4342. IEEE, Los Alamitos (2004)
6. Kruegel, C., Toth, T., Kirda, E.: Service specific anomaly detection for network intrusion detection. In: Proceedings of the 2002 ACM symposium on Applied computing, pp. 201–208. ACM Press, New York (2002)
7. Freund, Y., Schapire, R.: A decision theoretic generalization of on-line learning and an application to boosting. Comput. Syst. Sci. 57(1), 119–139 (1997)
8. Xu, L., Krzyzak, A., Suen, C.Y.: Methods of Combining Multiple Classifier and Their Application to Handwriting Recognition. IEEE transactions on systems, man and cybernetics 22(3) (May/June 1992)
9. Barbarra, D., Couto, J., Jajodia, S., Popyack, L., Wu, N.: ADAM: Detecting Intrusion by Data Mining. In: Proceedings of the 2001 IEEE, Workshop on Information Assurance and Security T1A3 1100 United States Military Academy, West Point, NY (June 2001)
10. Anderson, D., Frivold, T., Valdes, A.: Next-Generation Intrusion Detection Expert System (NIDES)-A Summary, Technical Report SRICLS-95-07, SRI (May 1995)
11. Zhang, J., Zulkernine, M.: A Hybrid Network Intrusion Detection Technique Using Random Forests. In: Proc. of the International Conference on Availability, Reliability and Security (AReS), Vienna, Austria, April 2006, pp. 262–269. IEEE Computer Soceity Press, Los Alamitos (2006)
12. Locasto, M., Wang, K., Keromytis, A., Stolfo, S.: Flips: Hybrid adaptive intrusion prevention. In: Valdes, A., Zamboni, D. (eds.) RAID 2005. LNCS, vol. 3858, pp. 82–101. Springer, Heidelberg (2006)
13. Porras, P.A., Neumann, P.G.: EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances. In: Proceedings of 20th National Information Systems Security Conference (1997)

14. Tombini, E., Debar, H., Mé, L., Ducassé, M.: A Serial Combination of Anomaly and Misuse IDSes Applied to HTTP Traffic. In: Proceedings of the Annual Computer Security Applications Conference (ACSAC) (December 2004)
15. Depren, O., Topallar, M., Anarim, E., Ciliz, M.K.: An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks. Expert Systems with Applications 29(4), 713–722 (2005); Elsevier
16. Hwang, K., Cai, M., Chen, Y., Qin, M.: Hybrid Intrusion Detection with Weighted Signature Generation over Anomalous Internet Episodes. IEEE Transaction on Dependable and Secure Computing 4(1), 41–55 (2007)
17. Xiang, C., Lim, S.M.: Design of Multiple-Level Hybrid Classifier for Intrusion Detection System. In: Proceeding of Machine Learning for Signal Processing, 2005 IEEE Workshop, pp. 117–122, September 28 (2005)
18. Rasoulifard, A., Ghaemi Bafghi, A.: Incremental Intrusion Detection Using Learn++ algorithm. In: 3rd conference on Information and Knowledge Technology, IKT2007, Ferdowsi University of Mashhad, Faculty of Engineering, November 27-29 (2007)
19. Yang, W., Yun, X.-C., Zhang, L.-J.: Using Incremental Learning Method From Adaptive Network Intrusion Detection. In: Proceedings of the Fourth International Conference on Machine Learnining and Cybernetics, Guanbzhou, August 18-21 (2005)
20. Zhong, S., Khoshgoftaar, T., Seliya, N.: Clustering-Based Network Intrusion Detection. International Journal of Reliability, Quality and Safety Engineering

A Cluster-Based Key Establishment Protocol for Wireless Mobile Ad Hoc Networks

Mohammad Sheikh Zefreh, Ali Fanian, Sayyed Mahdi Sajadieh, Pejman Khadivi, and Mehdi Berenjkoub

Department of Electrical and Computer Engineering
Isfahan University of Technology, Isfahan, Iran
{sheikhzefreh,fanian,sajadieh,pkhadivi}@ec.iut.ac.ir,
brnjkb@cc.iut.ac.ir

Abstract. Mobile Ad Hoc Networks (MANETs), due to their lack of physical infrastructures or centralized online authorities, pose a number of security challenges. Traditional network authentication solutions rely on centralized trusted third party servers or certificate authorities. However, ad hoc networks are infrastructure-less, and there is no centralized server for key establishment. Hence, traditional solutions do not meet the requirements of MANETs. In this paper, a key establishment protocol for mobile ad hoc networks is proposed which does not require any centralized support. The mechanism is built using the well-known technique of threshold secret sharing scheme and network clustering. This protocol is robust and secure against a collusion of up to a certain number of nodes and is well adapted with node movements. Due to use of clustering, the imposed overhead is very low compared with other similar key establishment protocols.

Keywords: Ad Hoc Networks, Key Establishment, Clustering, Secret Sharing.

1 Introduction

A Mobile Ad Hoc Network (MANET) is dynamically formed by a collection of mobile devices without employing any existing network infrastructure or centralized administration. Security is an important aspect for this type of networks. Same to the wired networks, there are five main attributes of security for a wireless ad hoc network: availability, confidentiality, integrity, authentication and non-repudiation. Availability is to ensure that the network services survive in the presence of denial of service attacks. A denial of service attack could be implemented at any layer of the ad hoc network. Confidentiality ensures that information is passed only to the authorized members of the network. Integrity is to guarantee that a message is transferred without getting corrupted. Authentication enables a node to identify the identity of the peer node it is communicating with. This is a useful property to detect adversary or compromised nodes. Non-repudiation is to ensure that a node can not deny having sent/received the message. There are many ways that an ad hoc network can be attacked upon. The use of wireless links gives ample opportunity for attacks ranging from passive eavesdropping to active ones.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 585–592, 2008.

Beyond the security issue, utilization of virtual backbones or clusters in mobile ad hoc networks has proven to be effective in solving several problems, such as minimizing the amount of storage for communication information e.g. routing and multicast tables, reducing information update overhead, optimizing the use of network bandwidth, service discovery, network management and etc. [1], [2].

In this paper, a key establishment protocol is proposed which is based on network clustering and secret sharing concepts. In the proposed protocol, in order to provide a fault tolerance key management protocol, clustering issue is considered. The proposed scheme is well adapted with node mobility and due to the use of clustering in the network, the imposed overhead is very low compared with some similar key establishment protocols. A collection of cluster heads (CHs) provides and simulates a Certificate Authority (CA) and based on secret splitting, similar to Shamir's scheme [3], it is sufficient for a cluster head to have interaction with K out of m ($K < m$) cluster heads (or secret holder) to reach CA's private key and by means of this key the CH can sign the identity of a newly joint node to its cluster. This node, by means of this signed identity (ID), can negotiate with a partner and authenticate him. Then, based on the proposed protocol, a shared key will be established. However, if a CH can reach CA's private key, it will become a security bottleneck and the meaning of secret sharing will be vanished. To avoid this problem, we propose partial signature of quorum of CHs in order to issue a valid certificate. In the sequel, by exploiting the proposed distributed CA, an efficient key establishment protocol for clustered ad hoc networks is proposed.

The rest of the paper is organized as follows: In Section 2, we discuss related works on key establishment over wireless ad hoc networks. Secret sharing scheme is described in Section 3. The proposed protocol is introduced and evaluated in Section 4. Section 5 is dedicated to conclusion.

2 Related Works

Recently, the problem of key establishment over wireless ad hoc networks has been studied. In [4], [5], [6], key pre-distribution has been discussed. Zhu et al. in [4] talk about a probabilistic key sharing scheme in which an off-line key server is used to initialize all of the nodes. In [6], multi-party key establishment based on password was proposed, where all the nodes are assumed to share a password. Basagni et al. in [5] explain a secure ad hoc network in which all the nodes split a group identification key, stored in tamper-resistant devices. Though all of the above schemes perform efficiently, they require that all the nodes have some pre-determined knowledge. In ad hoc networks, where mobile nodes do not have the privilege of knowing other group members beforehand, assumption of such a pre-shared secret is unrealistic. Traditional network authentication solutions rely on centralized trusted third-party servers, called certificate authorities (CAs). However, ad hoc networks are infrastructure-less, and there is no centralized server for key management. Hence, traditional solutions do not meet the requirements of mobile ad hoc networks. The mobile certificate authority concept has been discussed in [7], [8], [9]. In such schemes, the responsibilities of a CA are distributed among a set of wireless nodes. A subset (threshold) of such CAs must cooperate to obtain a valid certificate. Such schemes have several advantages

such as providing data integrity, authentication and non-repudiation. Also, several ID based key exchange protocols are proposed in [10, 11].

Our proposed protocol exploits the benefits of the pervious works, such as clustering and threshold cryptography, to introduce a suitable key establishment protocol for wireless mobile ad hoc networks.

3 Secret Sharing Scheme

In mobile ad hoc network environment, a single CA node could be a security bottleneck if it is not well protected. Multiple replica of CA are fault tolerant, but the network is as vulnerable as single CA or even worse since breaking one CA means breaking all CAs, meanwhile it could be much easier for attackers to locate a target. Hence, an elegant secret sharing scheme is widely proposed in mobile ad hoc networks with different implementations. To better understand this scheme, a short brief is introduced here.

The CA's private key, $K_{d,ca}$, which is a system-wide secret, is distributed to multiple nodes. No single node knows or can deduce the secret from the piece it holds. Only a threshold number of nodes can deduce the secret. In the next subsection we describe Shamir's secret sharing scheme and then explain application of this approach in the proposed key establishment between two entities in the network.

3.1 Shamir's Secret Sharing Scheme

In Shamir's (*k*,*n*) secret sharing scheme [3], a dealer shares a secret $K_{d,ca}$, CA's privet key, among *n* cluster heads such that :

1) Given any *k* shares, $K_{d,ca}$ could be recovered.

2) Given any $k_0 < k$ shares, nothing could be learned about $K_{d,ca}$.

The distribution of secret $K_{d,ca}$ is performed by a dealer as follows:

1. Select a large prime *P* and a random polynomial

$$f(x) = \alpha_0 + \alpha_1 x + ... + \alpha_{k-1} x^{k-1} \mod P \quad (1)$$

over Z_P of degree $k - 1$, satisfying $f(0) = K_{d,ca}$.

2. Give $x_i = f(i)$ to S_i (*i*'th share holder), $i = 1,2,...,n$.

When *k* cluster heads cooperate, the recovery of $K_{d,ca}$ is straightforward; Given (i, x_i) for $i \in \phi$ where $\phi \subset \{1,2,...,n\}$ satisfying $|\phi| = k$, $f(x)$ can be obtained as follows:

$$f(x) = \sum_{i \in \phi} \left(x_i \prod_{j \in \phi, j \neq i} \frac{x - j}{i - j} \right) = \sum_{i \in \phi} (x_i L_i^x)$$

Where

$$L_i^x = \prod_{j \in \phi, j \neq i} \frac{x - j}{i - j}$$

is the Lagrange coefficient. Therefore:

$$K_{d,ca} = f(0) = \sum_{i \in \phi} (x_i L_i^0) \quad [3].$$

In the proposed key establishment framework, $K_{d,ca}$ is distributed to n cluster heads based on the Shamir's secret sharing. Quorum of k ($1 < k \leq n$) CHs can produce a valid certificate.

4 Proposed Certificate Issuance and Key Establishment Protocol

In this section, according to Shamir's secret sharing, which described in previous section, a certificate issuance service and key establishment protocol is proposed. In order to reduce the overall network overhead, we use clustering structure and each cluster head is responsible for verifying user identity numbers and generating the corresponding certificate.

4.1 Assumptions

In the proposed protocol, we assume that:
p and q are large and strong primes.
n is the product of p and q.

$$\varphi(n) = (p - 1)(q - 1)$$

$$K_{d,ca} = g^S \quad Mod \quad \varphi(n)$$

S is a number which is selected by system dealer.

$$K_{e,ca} = K_{d,ca}^{-1} \; Mod \; \varphi(n)$$

(n , $K_{e,ca}$) is Certificate Authority's (CA) public key and $(S, p, q, K_{d,ca}, \varphi(n))$ is its private key.
(e_i , d_i) is public and private key of user i (U_i) respectively.

In addition, primitive element g in both of the fields *GF(p)* and *GF(q)* is determined, and a one-way hash function, H, is chosen. (g , H) along with (n, $K_{e,ca}$) and $\varphi(\varphi(n))$ are public information.

4.2 Proposed Certificate Issuance Service

Certificate of different users in the network, which is issued by means of a trusted third party, is a means for user's authentication. In this subsection proposed certificate issuance service is describe.

Suppose that user i (U_i) needs a new certificate. One approach is that he sends his identification number (ID_i) and public key (e_i) to its corresponding cluster head (CH_i) in order to obtain the signature $Sign_i$ which will be used as his certificate. If the cluster

head confirms the correctness of relationship between U_i and (e_i, ID_i), calculates $Sign_i$, using Shamir scheme as follows:

$$\mathrm{Sign_i} = (e_i, ID_i)^{K_{d,ca}} \quad Mod\ n$$

and hands $Sign_i$ to U_i[1]. (e_i, ID_i) indicates a one-to-one map of e_i and ID_i concatenation to Z_n . However, if a *CH* can reach *CA's* private key, it will become a security bottleneck and the meaning of secret sharing will be vanished. Thus, in order to avoid this problem we use the following approach:

Instead of distributing secret $K_{d,ca}$ among cluster heads, S is distributed by a dealer as follows:

1. He selects a random polynomial

$$f(x) = S + a_1 x + ... + a_{k-1} x^{k-1} \mod \varphi(\varphi(n))$$

2. Then, he gives $S_i = f(i)$ to cluster head i (i'th share holder), $i = 1,2,...,n$.

In the proposed protocol, without loss of generality, we assume that this secret sharing mechanism is verifiable. A secret sharing scheme is verifiable if sufficient auxiliary information is included that allows players to verify their shares as consistent. More formally, verifiable secret sharing ensures that even if the dealer is malicious, there is a well-defined secret that the players can later reconstruct [12]. Verifiable secret sharing issue is out of the scope of this paper.

When user i wants to gain a new certificate, he should give its initial certificate along with its identity and public key (ID_i ,e_i) to its corresponding cluster head (CH_i). The cluster head sends a request to other cluster heads and asks them to introduce themselves by sending their identities. (In this protocol we assume that there exist a secure relationship between CH_i and other cluster heads.). After gathering k identities, CH_i can produce the following equations (for simplicity, suppose that gathered identities are1, 2, 3,…, k):

$$\begin{cases} S_1 = f(1) = S + a_1 + ... + a_{k-1} & Mod \quad \varphi(\varphi(n)) \\ S_2 = f(2) = S + 2a_1 + ... + 2^{k-1} a_{k-1} & Mod \quad \varphi(\varphi(n)) \\ \vdots & \\ S_k = f(k) = S + ka_1 + ... + k^{k-1} a_{k-1} & Mod \quad \varphi(\varphi(n)) \end{cases}$$

Now, he must find coefficients A_1, A_2,…, A_k such that:

$$\begin{cases} A_1 S_1 = A_1 S + A_1 a_1 + ... + A_1 a_{k-1} & Mod \quad \varphi(\varphi(n)) \\ A_2 S_2 = A_2 S + 2A_2 a_1 + ... + 2^{k-1} A_1 a_{k-1} & Mod \quad \varphi(\varphi(n)) \\ \vdots & \\ A_k S_k = A_k S + A_k k a_1 + ... + A_k k^{k-1} a_{k-1} & Mod \quad \varphi(\varphi(n)) \end{cases}$$

[1] In this protocol, it is assume that each cluster head can authenticate user i and vice versa.

Summing these equations we have:

$$\sum_{i=1}^{k} A_i S_i = (\sum_{i=1}^{k} A_i)S + (\sum_{i=1}^{k} A_i i)a_1 + \ldots + \sum_{i=1}^{k}(A_i i^{k-1})a_{k-1} \quad Mod \quad \varphi(\varphi(n))$$

These coefficients can be found by the following equations:

$$\begin{cases} \sum_{i=1}^{k} A_i = 1 & Mod \quad \varphi(\varphi(n)) \\ \sum_{i=1}^{k} iA_i = 0 & Mod \quad \varphi(\varphi(n)) \\ \vdots & \\ \sum_{i=1}^{k} i^{k-1}A_i = 0 & Mod \quad \varphi(\varphi(n)) \end{cases}$$

Note that coefficients (a_1, a_2,…, a_{k-1}) and share of each user, S_i ,are unknown for CH_i. Basically, CH_i does not required to know those coefficients and shares. It is just needed to know which cluster head is available in the network.

After obtaining coefficients A_1,A_2,…,A_k , CH_i sends each coefficient to corresponding cluster head (for example, he sends coefficient A_1 to cluster head 1.) and asks the first cluster head, for example cluster head 1, to sign (e_i, ID_i) partially. On receiving these information, the first cluster head signs (e_i, ID_i) as follows (suppose that the first cluster head is CH_1):

$$(e_i, ID_i)^{g^{S_1 A_1}} \quad Mod \quad n$$

and gives his partial signature to second cluster head and asks him to sign it partially. Therefore, the second cluster head signs it as follows (suppose that the second cluster head is CH_2):

$$((e_i, ID_i)^{g^{S_1 A_1}})^{g^{S_2 A_2}} = (e_i, ID_i)^{g^{S_1 A_1 + S_2 A_2}} \quad Mod \quad n$$

and gives his partial signature to third cluster head an asks him to sign it partially. This process continues until that the last cluster head signs received signature partially and hands it to CH_i . At this time, complete signature on (e_i, ID_i) is obtained:

$$\begin{aligned} \text{Sign}_i &= (e_i, ID_i)^{g^{S_1 A_1 + S_2 A_2 + \ldots + S_k A_k}} && Mod\, n \\ &= (e_i, ID_i)^{g^{\sum_{j=1}^{k} A_j S_j \quad Mod \quad \varphi(\varphi(n))}} && Mod\, n \\ &= (e_i, ID_i)^{g^{S} \quad Mod \quad \varphi(n)} && Mod\, n \\ &= (e_i, ID_i)^{K_{d,ca}} && Mod\, n \end{aligned}$$

Thus, in this protocol, (e_i, ID_i) is signed and *CA*'s private key has not been disclosed.

One of the advantages of the proposed protocol is that this protocol supports dynamic nature of ad hoc networking and nodes' mobility. In fact, a virtual infrastructure always exists in the network and each node inside the network is associated with a *CH*. Therefore, that node can gain a new certificate or update its initial certificate using its corresponding CH.

4.3 Proposed Key Establishment Protocol

Suppose that U_i and U_j are the two users who want to communicate with each other. First, U_i selects a random number r_i and computes SK_i which is its semi-key:

$$SK_i = g^{r_i} \quad Mod\ n$$

Second, U_i uses a nonce N_i, identification number of user *j*, ID_j, and his public key to perform the operation of one way hash function $H(N_i, SK_i, ID_j, e_i)$, and then signs it by its private key. Hence, let us define:

$$h_i = H(N_i, SK_i, ID_j, e_i)\ ,\ Sh_i = h_i^{d_i} \quad Mod \quad n$$

Finally, U_i sends $(SK_i, Sh_i, N_i, ID_i, \mathrm{Sign_i})$ to U_j. Similarly, U_j selects the random number r_j and nonce N_j. Then computes

$$SK_j = g^{r_j} \quad Mod n$$
$$h_j = H(N_j \oplus N_i, SK_j, ID_i, e_j)$$
$$Sh_j = h_j^{d_j} \quad Mod \quad n$$

and sends $(SK_j, Sh_j, N_j, ID_j, \mathrm{Sign_j})$ to U_i.

Using the combination $N_i \oplus N_j$, it will be possible for user *i* to be confirmed of the aliveness of user *j* and re-play attack will be prevented.

Before generating the session key, U_j and U_i need to verify the integrity of $(SK_i, Sh_i, N_i, ID_i, \mathrm{Sign_i})$ and $(SK_j, Sh_j, N_j, ID_j, \mathrm{Sign_j})$. This is performed by evaluating the correctness and trustiness of the *CA*'s signature on the partner identity. This operation is done by U_j and U_i as follows (signature decryption by means of *CA*'s public key):

$$(e_i, ID_i) \stackrel{?}{=} Sign_i^{K_{e,ca}} \quad Mod \quad n\ ,\ (e_j, ID_j) \stackrel{?}{=} Sign_j^{K_{e,ca}} \quad Mod \quad n$$

After this verification, they extract partner's public key from the signature. Then U_j and U_i compute h_i and h_j from the received parameters and compare it with decrypted Sh_i and Sh_j in the following manner respectively:

$$h_i \stackrel{?}{=} \mathrm{Sh_i^{e_i}} \quad Mod \quad n\ ,\ h_j \stackrel{?}{=} \mathrm{Sh_j^{e_j}} \quad Mod \quad n$$

Note that in this protocol, users do not need to have previous knowledge of partner's public key, so they don't need any memory for public key maintenance.

Finally, U_i, and U_j compute session key *K*, as follows:

$$K = g^{r_i . r_j} \quad Mod \quad n$$

Due to use of nonce, identification information and one way hash function *H*, this key establishment protocol is survivable against *man in the middle attack*.

5 Conclusion

In this paper, a certificate authority service and key establishment protocol for ad hoc networks is proposed that its architectural framework enables the secure and dynamic use of services in wireless ad hoc networks. It has two foundations: network clustering which provides a virtual backbone, and secret sharing scheme. Due to the use of clustering structure, this protocol is well adapted with node mobility and the imposed overhead is very low compared with some similar key establishment protocols.

Acknowledgment

The authors would like to thank Rostam Shirani for his helpful comments and several instructive discussions.

References

1. Chatterjee, M., Das, S.K., Turgut, D.: WCA: A Weighted Clustering Algorithm for Mobile Ad hoc Networks. Journal of Clustering Computing IEEE 5(2), 193–204 (2002)
2. El-Bazzal, Z.: An efficient management algorithm for clustering in mobile ad hoc network. In: Proceedings of the ACM inter. workshop on Performance monitoring, measurement, and evaluation of heterogeneous wireless and wired networks (2006)
3. Shamir, A.: How to Share a Secret. Communications of the ACM 22, 612–613 (1979)
4. Zhu, S., Xu, S., Setia, S., Jajodia, S.: Establishing pair-wise keys for secure communication in ad-hoc networks: A probabilistic approach. In: IEEE International Conference on Network Protocols (2003)
5. Basagni, S., Herrin, K., Rosti, E., Bruschi, D.: Secure pebblenets. In: ACM Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc (2001)
6. Asokan, N., Ginzboorg, P.: Key establishment in ad hoc networks. Computer Comm. 23, 1627–1637 (2000)
7. Yi, S., Kraverts, R.: Key management in heterogeneous ad hoc wireless networks. In: IEEE ICNP (2002)
8. Kong, J., Zerfos, P., Luo, H., Lu, S., Zhang, L.: Providing robust and ubiquitous security support for mobile ad-hoc networks. In: IEEE ICNP (2001)
9. Wu, B., Wu, J., Fernandez, E., Magliveras, S.: Secure and efficient key management in mobile ad hoc networks. In: Proceedings of 19th IEEE International Parallel and Distributed Processing Symposium (2005)
10. Khalili, A., Katz, J., Arbaugh, W.: Toward secure key distribution in truly ad hoc networks. In: 2003 Symposium on Applications and the Internet Workshop (SAINT 2003), pp. 342–346. IEEE Computer Society, Los Alamitos (2003)
11. Hoeper, K., Gong, G.: Models of authentications in ad hoc networks and their related network properties. International Association for Cryptologic Research (2004)
12. Yang, H., Kim, J., Kwon, C., Won, D.: Verifiable secret sharing and multiparty protocols in distributed systems. In: ACM Symposium on Theory of Computing Proceedings (1989)

Time Series Analysis for ARP Anomaly Detection: A Combinatorial Network-Based Approach Using Multivariate and Mean-Variance Algorithms

Yasser Yasami, Saadat Pourmozaffari, and Siavash Khorsandi

Department of Computer Engineering,
Amirkabir University of Technology, Tehran, Iran
{yasami,saadat,khorsand}@ce.aut.ac.ir

Abstract. This paper presents a novel network-based ARP anomaly detection technique. The proposed approach applies a combinatorial algorithm composed of multivariate and mean-variance algorithms. The paper main objective is to construct a statistical ARP anomaly detection system capable of classifying the ensemble network ARP traffic as normal or abnormal. For this purpose time series data of normal purred ARP traffic from the ensemble network are analyzed and some statistical parameters are extracted. The observed time series data of ARP traffic parameters are compared by the normal parameters. Any deviation from the normal model is intended to be abnormal. The proposed technique is novel and this paper is the first publication that introduces a high accurate and performance Network-based ARP anomaly detection technique.

Keywords: Anomaly Detection System (ADS), Address Resolution Protocol (ARP), Time Series Data, Multivariate and Mean-Variance Algorithms.

1 Introduction

Traditional signature-based intrusion detection techniques use patterns of well-known attacks or weak spots of the system to match and identify known intrusions. The main drawback of theses techniques is inability to detect the newly invented attacks. Common techniques used to achieve acceptable information about network traffic and compensate this incompleteness of traditional IDS'es apply *anomaly detection algorithms* (ADA). Theses algorithms can be employed as useful mechanisms to analyze network anomalies and detect ill-actions issued by users, or even unknown signature viruses and worms that suppress network normal traffic [5], [6].

There are two main approaches to studying or characterizing the ensemble behavior of the network [7]: the first is inference of the overall network behavior and the second by understanding the behavior of the individual entities or nodes. The method presented in this paper deal with the ensemble network as a single entity and is categorized in the former approach.

Any protocol in the different layers of network can be affected by undesired factors and cause anomaly in the network traffic. One widely used protocol is *Address Resolution Protocol* (ARP). ARP is a broadcast protocol and has a simple structure, but it is potentially a harmful agent, if applied by malicious softwares.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 593–600, 2008.

The goal of this paper is to apply a combinatorial algorithm composed of multivariate and mean-variance algorithms to the problem of ARP anomaly detection. The proposed method uses multivariate algorithms criteria for purring time series data of captured ARP traffic. A mean-variance algorithm is deployed for extracting some statistical parameters. Anomaly detection is performed by comparison of normal parameters of purred ARP traffic time series data by parameters of observed ARP traffic time series data. Any deviation observed in this process is marked as abnormal.

After this introduction the rest of the paper is organized as follow: Section 2 describes background and related works. Section 3 surveys theory of algorithm including multivariate algorithm parameters and ARP traffic purring, modeling the normal ARP traffic, matching process and anomaly detection. In section 4 experimental results and evaluation of the proposed method is presented. Finally, the two last sections include conclusion and references.

2 Background and Related Works

Network anomaly detection is a vibrant research area and many researchers have approached this problem using various techniques [2], [9], [1], [4]. Some methods for anomaly detection are based on switch characteristics [9]. In such methods switch characteristics must be known. Our knowledge is limited to theoretical backplane speed mentioned in datasheets. But, because performance of switches in high load small packet traffic degrade dramatically [10], so using such algorithms, encounters functional limitations, making them unusable.

In the other researches [1], [4], a number of criteria for analyzing of ARP traffic have been suggested. Such methods are based on weighted summation of different ARP traffic criteria. To achieve more accuracy on the result, we need to define more factors as inputs to these algorithms. Furthermore, the proposed factors have correlation with each other. None of these references include any suggestion about correlation between the factors. So, such approaches have not enough precision.

Authors in [11] aim at classifying the TCP network traffic as an attack or normal. The main objective in [11] is to build an anomaly detection system for TCP connections and does not include any discussion about anomaly issues of layer two broadcast traffic, which is applied by some malicious softwares. So, this approach is unable in detecting scanning worms.

The method we proposed in [2] uses a markovian process for host-based anomaly detection and has computational overhead which limits its application for real network environments, practically. Our recent work [3] presents a host-based ARP anomaly detection technique and does not have network-based theory.

Furthermore, some of the mentioned methods are protocol-based which limits their generalization to ADA's based on other protocols, commands or user actions. There is not any feasible network-based ARP anomaly detection approach in open literatures, with acceptable precision and performance (to our knowledge).

3 Theory of Algorithm

Network anomalies typically refer to circumstances when network operations deviate from normal network behavior and can arise due to various causes [2]. The definition of normal network behavior for measured network data is dependent on several specific network factors such as the dynamics of network in terms of traffic volume, the type of available network data, and types of applications running on the network.

Some of intrusions and malicious usages don't have any significant effects on network traffic (i.e. ARP Spoofing). So such misbehavior is not addressed in this paper. Another type of attacks is based on broadcasting ARP packets with abnormal behavior (such as DoS attacks). Abnormality is generally different from a large number of ARP requests. There are other types of attacks which apply ARP for detecting live hosts in network. In addition to layer two origins of anomalies, any higher layer traffic anomalies affects on ARP traffic. ARP anomalies can be caused by some unintentional and curious motivations, too. To detect these anomalies an algorithm is introduced which has security and performance saving importance.

3.1 ARP Traffic Purring, Modeling and Extracting Statistical Parameters

Some anomaly criteria are defined indicating abnormal behavior of a host, for traffic purring. These criteria are basically the one's used in multivariate anomaly detection algorithms. ARP requests matching these criteria are intended to be abnormal. The main criteria are ARP rate, burstiness, sequential scans and dark space [3].

The proposed algorithm performs time series data analysis of purred ARP traffic of total hosts in the same broadcast domain. This process is referred to as training process. In this analysis the time series data is intended be composed of some states and transitions among them. A time series data of captured ARP traffic from a real network, and its corresponding states and transitions among them within a time interval of above 3 minutes and its corresponding state diagram is presented in figures (1) and (2), respectively. Each egregious change in average ARP requests per second in a state causes a state transition. Four statistical parameters are extracted from the time series data during training process, for each state, which are as follow:

- State Identity: Each state is identified by a main characteristic of "Average ARP Requests" which has limited variation during the occurrence of each state.
- State Probability: Probability of state S (PS) is defined as the number of occurrence of S, divided by total number of occurrence of all observed states:

$$P_S = \frac{n_S}{\sum_{i=1}^{k} n_i} \tag{1}$$

Where, n_i is the number of occurrence of the i, and k is the whole number of observed states in the time series data in training process.

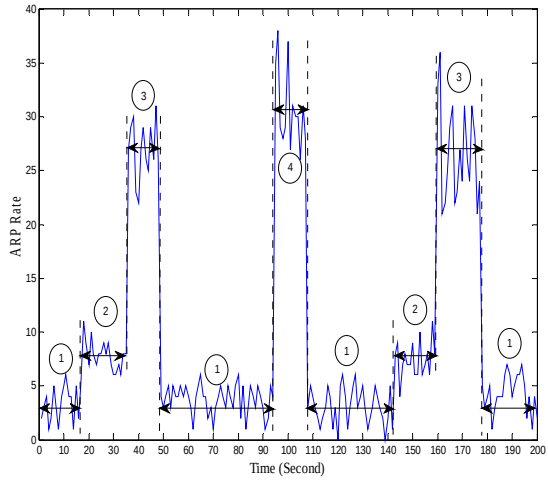


Fig. 1. A time series data of ARP traffic and its corresponding states and transitions

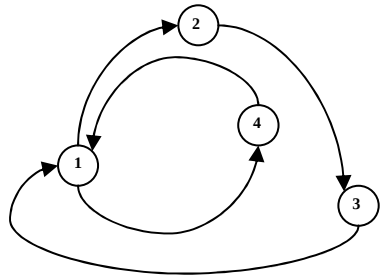


Fig. 2. The corresponding state diagram of the above time series data

- *Steady State Duration Average*: Steady State Duration Average of state S (ε_S) is the mean time that the time series reside in state S without any state change:

$$\varepsilon_S = \frac{\sum_{i=1}^{n_S} t_{Si}}{n_S} \tag{2}$$

Where, t_{S_i} is the time interval that i[th] observation of state S last. The denominator statement is already described above.

- *Steady State Duration Variance*: This parameter for each state is a measure of ARP rate variation in that state. $E[t_S]$ and $E[t_s^2]$ are expected values.

$$\sigma_S^2 = E[t_S^2] - E[t_S]^2 \tag{3}$$

3.2 Matching Process, Anomaly Threshold and Anomaly Detection

The process of comparing network traffic by the normal model is referred to as *matching process*. In this process, the whole ARP traffic is compared by its normal model, and any deviation from normal model affects on *anomaly score* (*AS*). *AS* is an anomaly indicator for any decision on network abnormality. One important factor for this decision is a threshold value. An indicator of normality in training process, called *Normal Score* (*NS*) is used for this purpose. It is described latter.

Anomaly Score (*AS*). We define *AS* as an anomaly indicator of ARP traffic and obtain it from weighted summation of *Partial Anomaly Scores* (*PAS*'es) as follow:

$$AS = K_s A_s + \sum_{k=2}^{N} \frac{K_j A_j^k}{P_{ij}} \tag{4}$$

Where N is the number of times network changes its state, j is the state in the matching process which the network will be in after kth time network changes its state and A_j^k is the *PAS* and is explained latter.

A_S is the initial *PAS* and corresponds to the state which the network will be inside as the matching process starts and can be stated formally as $A_S = A_i^1$, for each i. K_j is coefficient of the participating term in weighted summation *AS* and is inversely dependent on probability of state j. So, inverted state probability is used for this parameter. This justification can be described for transition probabilities.

P_{ij} is conditional probability of transition from state i to state j caused by k^{th} change in network state, given the sequence of observed transitions in matching process.

$$P_{ij} = P(T_{ij} \left| T_{I_1 I_2} T_{I_2 I_3} ... T_{I_{k-2} I_{k-1}} \right.) \tag{5}$$

Where, T_{ij} is transition from state i to state j, $T_{I_1 I_2} T_{I_2 I_3} ... T_{I_{K-2} I_{k-1}}$ is sequence of observed transitions in matching process, I_k indexes correspond to states where the network will be in after kth state change in matching process, as I_1 and I_{k-1} Indexes correspond to states S and i, respectively. This conditional probability is as follow:

$$P_{ij} = \frac{P(T_{I_1 I_2} T_{I_2 I_3} ... T_{I_{k-2} i} T_{ij})}{P(T_{I_1 I_2} T_{I_2 I_3} ... T_{I_{k-2} i})} \tag{6}$$

Partial Anomaly Score (*PAS*). Defined as deviation from average steady state duration in a state and is calculated as (7). t_{jk} is time interval between k^{th} and $(k+1)^{th}$ transition in matching process (k^{th} transition leads to change to state j).

$$A_j^k = \frac{(\varepsilon_j - t_{jk})^2}{\sigma_j^2} \tag{7}$$

One problem is about entities (states and sequences of transitions) which are not observed in the time series data. Minimum state and transition probability among the whole probabilities are taken for these probabilities. Also, we considered statistical parameters of each state X not involved in the trained model (σ_X^2 and ε_X) for every state i in the model, as $\varepsilon_X = MAX\{\varepsilon_i\}$, $\sigma_X^2 = MIN\{\sigma_i^2\}$. It means taking worst case.

Normal Score (*NS*). *NS*, is an indicator of normality degree and depends on partial *NS*'es (*PNS*). PNS_i is defined as normality score in i^{th} observation interval in training process. *NS* calculates as $NS = MAX\{PNS_i\}$. The same method as used for calculation of *AS* is applied about PNS_i, but in this case using normal ARP traffic.

Selection of threshold value affects on the number of *False Positive* (FP) and *False Negative* (FN) alarms. An estimation of ARP traffic normality is required for estimating a right threshold value. The *NS* get this estimation. The computed value of *AS* for any matching process should satisfy the inequality $AS \geq Th > NS$ to be detected as abnormal. Where, *Th* is the threshold value. A simple and feasible way for calculating threshold is to set threshold value k times of *NS*, as is applied by [8].

4 Evaluation and Experimental Results

Our test bed network in this research work was a typical network with about 900 active devices. With exception of a few Linux and Novell servers, all of hosts run different Microsoft Windows platforms. The network is connected to Internet and about 200 stations and 5 servers are connected to internet concurrently.

A computer connected to the core switch of the network is used to capture traffic. Capturing traffic and some statistical parameters from it, is performed in real-time interaction with our prototype, by setting the sniffing machine NIC in sniffing and statistical mode. WinPcap API has been used for this purpose.

As stated above the initial phase in anomaly detection is filtering ARP traffic from abnormalities, according to anomaly criteria described above, in section 3.1. We have presented that these anomaly criteria can give very good description of abnormalities in network ARP traffic and are very effective in purring traffic and providing normal condition which the normal model is built on [3].

The main advantage of the proposed approach is its high performance and precision in anomaly detection. Its high performance is because of its inherent nature which behaves with the whole network as a unique identity and detects any misbehavior in ARP traffic, regardless of its source. This causes the number of states which the network can be inside, very little.

Of course, the number of states affects on the precision of the proposed algorithm as well as its performance. The main parameter which drives total number of states is variance of ARP requests over different time intervals (with same length) during each observed state, as stated above in section 3.1, about *State Identity*. Choosing little values for this parameter causes large number of states which impose extra computational overhead of dealing with large number of states. Beside, large values of this parameter decrease the number of states and therefore lower algorithm accuracy. However, very large number of states does not imply very high accuracy.

So, this parameter has a significant impact on algorithm results. Its value is dependant on the number of existing nodes in the network and network activity. Network activity depends on type of network applications, user activities and etc. ARP rate variance calculated over total time series data can give an estimation for choosing a right value for this parameter. For our network test bed an estimation of about 87.65 (standard deviation of about 9.36) has been calculated for this parameter.

The experimental results of the proposed approach in the real campus network show the time series data of network ARP traffic is composed of some limited number of states and transition among them, where ARP rate has an acceptable variation in each state compared by state identity (state ARP rate). This is the main reason that our proposed approach is a practical and feasible technique.

Our evaluation is based on running our prototype in the real network and measuring the number of FP alarms. The result shows that the number of FP is in an acceptable range as the table (1) indicates.

As this table indicates, network scan tools and scanning worms by issuing a large number of ARP packets in a little time interval have a large influence on ARP rate and so ARP abnormalities. They assigned a large percentage of alarms to themselves. Network applications, user activities and other origins of abnormalities (such as malfunctioning or bad configured network devices, other malicious softwares and etc.) affect on ARP traffic and issued abnormalities to it but had a less portion.

Table 1. Evaluation results of the proposed algorithm within four weeks

Week	Number of Alarms	Number of FP's	Anomaly Origins		
			Scan Tools & Scanning Worms	Network Applications & User Activities	Others
1	16	1	12	2	1
2	21	2	14	4	1
3	11	1	8	2	0
4	15	1	12	1	1
Percentage	92%		73%	14.30%	4.70%

5 Conclusion

In this paper we studied anomaly detection in ARP traffic behavior and presented a novel network-based ARP ADA by analyzing the ensemble network ARP traffic. The proposed technique performs time series data analyzing of normal ARP traffic and extract some statistical parameters from it. The statistical parameters of observed time series data are compared by the normal parameters. Traffic purring is performed based on some defined anomaly criteria. The main objective of this paper was to build a predictive model capable of detecting anomaly in network ARP traffic.

Experimental results of the proposed technique in a real campus network showed that this technique is powerful and accurate. Furthermore, because it behaves with the total network ARP traffic as a single entity, regardless of traffic sources, the model has a little number of states and transitions among them, which this leads to very low computational overhead. The number of FP alarms was used as an indication of accuracy which was in an acceptable range, as the experimental results show.

References

1. Whyte, D., Kranakis, E., Van Oorschot, P.: ARP-Based Detection of Scanning Worms within an Enterprise Network. In: Annual Computer Security Applications Conference (ACSAC 2005), Tucson, AZ (2005)
2. Yasami, Y., Farahmand, M., Zargari, V.: An ARP-based Anomaly Detection Algorithm Using Hidden Markov Model in Enterprise Networks. In: 2nd International Conference on Systems and Networks Communication (ICSNC 2007), France, p. 69. IEEE, Los Alamitos (2007)
3. Yasami, Y., Pourmozaffari, S., Khorsand, S.: An ARP-based combinatorial approach based on mean-variance & multivariate algorithms for anomaly detection in enterprise networks. In: 3rd Int'l conference of Information & Knowledge Technology, IKT 2007 (2007)
4. Farahmand, M., Azarfar, A., Jafari, A., Zargari, V.: A Multivariate Adaptive Method for Detecting ARP Anomaly in Local Area Networks. In: International Conference on Systems and Networks Communication (ICSNC 2006), pp. 53–59. IEEE, Los Alamitos (2006)
5. Maselli, G., Deri, L.: Design and Implementation of an Anomaly Detection System: an Empirical Approach. In: Terena TNC 2003, Zagreb, Croatia (2003)
6. Hwang, K., Liu, H., Chen, Y.: Protecting Network-Centric Systems with Joint Anomaly and Intrusion Detection over Internet Episodes. In: IEEE IPDPS- 2005 (2004)
7. Thottan, M., Ji, C.: Anomaly Detection in IP Networks. IEEE Trans. Signal Processing 51(8) (2003)
8. Hwang, K., Liu, H., Chen, Y.: Cooperative Anomaly and Intrusion Detection for Alert Correlation in Networked Computing Systems. IEEE Transaction on Dependable and Secure Computing (2004)
9. Ármannsson, D., Hjálmtýsson, G., Smith, P.D., Mathy, L.: Controlling the Effects of Anomalous ARP Behaviour on Ethernet Networks. In: 2005 ACM conference on Emerging network experiment and technology, pp. 50–60 (2005)
10. Cisco Systems Catalyst 6500 vs. Foundry Networks BigIron 8000.: Summary Test Report Core Ethernet Switches Buffering and Control Plane Performance Comparison. MIER Communications Inc. (2000)
11. Joshi, S.S., Phoha, V.V.: Investigating Hidden Markov Models Capabilities in Anomaly Detection. In: 43rd ACM Southeast Conference, Kennesaw, GA, USA (2005)

Polyhedral GPU Accelerated Shape from Silhouette

Alireza Haghshenas[1], Mahmoud Fathy[1], and Maryam Mokhtari[2]

[1] Department of Computer Engineering
Iran University of Science and Technology, Tehran, Iran
Haghshenas@comp.iust.ac.ir, Mahfathy@iust.ac.ir
[2] Department of Computer Engineering
Isfahan University of Technology, Esfahan, Iran
Mokhtari@ec.iut.ac.ir

Abstract. A fast method for shape from silhouette is proposed with adjustable accuracy. This method uses polyhedral representation of the shape and therefore needs much less memory than traditional voxel based approaches. The proposed method also makes use of traditional GPUs processing power to accelerate the recovery process and free up some CPU cycles for other tasks. The output polyhedron has large non-fragmented faces facilitating reflection-based analyses for further shape refinements.

Keywords: Shape from Silhouette, Polyhedron, GPU Acceleration, Polyhedron Clipping, Polyhedron Intersection.

1 Introduction

Recent advances in computer graphics, has triggered a demand on more and better 3D models to be available for several application areas. Subject to this demand, traditional methods of creating graphical models, i.e. Expert human modelers using 3D modeling packages are rapidly becoming obsolete due to time, cost and quality. Even using 3D laser scanners to obtain the shape of existing objects fails due to poor alignment of the texture and shape. A better approach is to use several images from ordinary cameras to recover the 3D model, similar to what happens in human brain.

Typically, one has to reduce the generality of the problem, and solve some limited version. Any knowledge about the scene can reduce the complexity: assuming lambertian reflectance, knowing the relative position of object to camera and/or light sources, having prior knowledge of the object (i.e. its convexity, smoothness and being highly textured).

There are still some areas in which such simplifications are not valid; recovering car shapes is an example: they principally lack textures, they exhibit specular reflectance, it's much harder to control the relative position of camera and object for such big objects and, the prior knowledge of their shape is sparse.

2 Related Work

In [1, 2, 3], a dense stereo matching is applied between each pair of neighboring images and the depth is calculated from each pair. The extracted depth information,

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 601–608, 2008.

together with automatic camera calibration based on an extension of the 8-point algorithm and a robust outlier rejection scheme, forms a completely autonomous shape recovery infrastructure. Their system performs great on highly textured, lambertian surfaces, but the usage of the dense stereo matching step distorts the result on large untextured areas and on non-lambertian surfaces.

Space carving algorithm [4] and its descendants work on a cubic grid of small 3d units called voxel. Based on a well axiomatic discussion introduced in the original paper, a voxel violating the color consistency criteria is known to be excluded from any possible photo-consistent volume and therefore is removed from the set. The final remaining volume is "almost" guaranteed to be the union of all photo-consistent volumes. The problem with this algorithm is twofold; first, it causes random outliers on randomly consistent voxels, specially in less textured areas. Second, the original color consistency criterion only performs well on lambertian surfaces and any noticeable specular reflection will reduce the result, possibly to an empty set.

The improved color-consistency scheme proposed in [5] partially solves the problem with specular reflections but still fails where there is little diffuse texture and it introduces additional probability for outliers, as its consistency condition is in fact a more relaxed version of the original one.

Another approach, not inherently complete in nature, is shape from silhouette [6]. Each 2D silhouette (i.e. from each image) confines the 3D object to an infinite conic-like shape, defined by the camera's center of the projection and the borders of the silhouette. The intersections of all these conics is a rather convexed approximation of the actual shape, hence called the visual hull. It can be used as a starting point for further carvings through stereo matching or photo-consistency carving methods.

Another refinement method is proposed in [7], in which the hall voxels are approximated by a triangulated mesh and then the mesh is optimized to minimize a cost function of color-inconsistency, outward silhouette inconsistency and shape roughness. The optimization is done in iterative single-node movements.

In contrast to the voxel based methods, a number of polygon based approaches have been proposed. These methods will be discussed in section 3.1.

3 Polyhedron Based Shape from Silhouette

In this paper, a polyhedron based shape from silhouette scheme is proposed. The idea behind this approach is simple: instead of representing the volumes by voxel collections, we use a polyhedron to represent a given 3d shape. In particular, our method uses meshes, which are face-triangulated polyhedra suitable for graphics representations. In the 2d steps, instead of handling the silhouette obtained from each image as a collection of pixels, they have to first be approximated by polygons to the desired precision level. This process can be done in an "iterative refinement" in which the approximation is refined by splitting edges, or in a "progressive" manner where an edge is extended from the starting point until no further extension is possible.

In general, none of the two schemes are optimal, i.e. none can guarantee to find the minimum number of edges for a given distance threshold. An optimal method is given in [8], restricted to discrete curves, i.e. curves on an integer grid. In most cases, this limitation causes no problem; however, if sub-pixel accuracy is to be used, a

general curve approximation of one of the sub-optimal schemata has to be used. From the resulting polygon, an infinite conic like polyhedron can be constructed in which the final shape is expected to reside. Any of the faces of this polyhedron (except the initial polygon) is a triangle. In order to be able to present this conic to a graphics device, we will have to triangulate its polygonal face using Delaunay triangulation.

3.1 Polyhedral Intersection

The next step would be to intersect all these conic like polyhedra in order to obtain the visual hull. Efficient polyhedral intersection methods offered in computational geometry literature are restricted to convex polyhedra. A good example is [18], in which a linear-time algorithm is shown to exist that makes use of a k-shield geometric data structure to keep track of the location in a polyhedron in logarithmic time. Unfortunately, the convexity is an inherent requirement for the construction and proper operation of the k-shield. An important movement to handle more general problems is [19] in which the convexity criterion is relaxed from one of the polyhedra. By the way, some methods exist in computer vision literature to handle the general polyhedral intersection. One is the "Marching Intersections", proposed in [11], in which the intersection problem is reduced to intersections in small cubic cells, each one being completely a member of one polygon, or completely out of the polygon, or being cut with a small plane. The intersections problem is handled in each cube separately and a complete model is obtained. Their method needed a much coarser grid than the voxel-based methods to exhibit the same level of precision; therefore it can run much faster; By the way, the output shapes will have very fragmented faces, that totally prevents effective specular analysis on them, for our further refinements. Also, these fragmented faces are much slower to be presented to display systems for simulations. Finally, the result of shading them will look rugged, because the normal of small nearby faces differ by large amounts. Finally, a real polyhedral approach is proposed in [12], where some observations are used to create a fast algorithm. These observations are: each clipping conic is the projection of a fixed cross-section onto space, and each of its faces is a triangle. They projected each triangle of one of the conics on all other image planes; the result of this projection, except in degenerate cases is still triangle. Then, they intersect this triangle with the other silhouettes, and project the intersection back to the space. They use an edge-bin data structure to speed up their calculations.

3.2 Our Method

A common ability of graphics processors is to tell if a given ray meets a given mesh representation of an object, and if it does, where and to what faces. This ability is used in several functions of DirectX™ as in "Intersect" function. It is used for collision detection, for ray tracing, and shading.

Our solution to the mesh intersection problem is to use a similar method introduced in [19], but instead of using a k-shield representation, we use the "Intersect" function to determine all collisions of all edges of one polyhedron with the other one. Let the polyhedra to be clipped be named P and Q. It can be observed that when the two polyhedra are in general position (i.e. no vertex of one resides on a face of the other one and no edge or face of one is coplanar with an edge or face of the other),

- Any vertex of the intersection is the intersection of one vertex of P with one face of Q, or one vertex of Q with one face of P.
- Any edge of the intersection is the intersection of one face from P and one face from Q.
- Any face of the intersection will be a clipped part of one face of P to Q or one face of Q clipped to P.

In order to keep the conic sections look solid, and for the clipping algorithm to be able to discriminate between filled and hole parts in the silhouette, we have used the culling convention; so one has to keep in mind to traverse the triangles of the holes in opposite order, so the face normal will always face outside the volume. This way, the intersection algorithm can know which side of any face is "inside" just by looking at its normal and no global calculations are required.

Suppose we are currently on vertex v_1 of the resulting polyhedron, called R. Also, let v_1 be the intersection of edge e_1 of face f_1 of P, and face f_2 of Q. when moving along face f_1, there are two possibilities: either another one of f_1's edges, e_2, intersects f_2, or one of f_2 edges, e_3, intersects f_1. These two cases are depicted in **Fig.** .

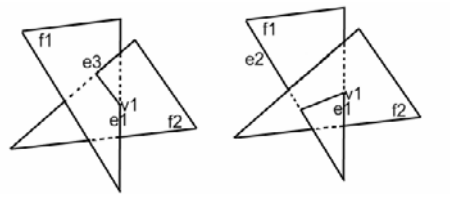


Fig. 1. Two possibilities when two triangular faces intersect

In order to be able to determine these states efficiently, the outputs of "Intersect" operations should be stored in a data structure similar to adjacency lists in graph algorithms; for each face of P we store what edges of Q intersect it, and where the intersection happens. A more detailed discussion reveals that even when the polyhedra are not in general position, the same adjacency representation is still valid.

The problem can now be looked at in this way: for each face of one polyhedron, say P, we know in a list, what edges of the same polyhedron (P) are making it, also we know in another list, what edges of the other polyhedron (Q) intersect it. The locations of the intersection are also known, the problem is now to define edges of the intersection polyhedron (R), i.e. to connect vertices of it in a way that any two subsequent vertices are on the same face of P and on the same face of Q.

Vertices making R polyhedron are: Points of P inside Q, Points of Q inside P and the intersections of P edges with Q faces and vice versa.

As it turns out, there are no points of P inside Q; remember that P is in fact a limited cut of an infinite conic like volume. The center of the projection is the camera location, outside the final shape. Also, its other vertices are in an arbitrary distance of the center that, to create correct results has to be certainly farther than any point of the actual object. If after the first intersection the remaining volume is bounded, no problem will be thereafter. For the first intersection, we have to choose two images nearest to perpendicular, so they will not have an infinite volume in common (**Fig.**).

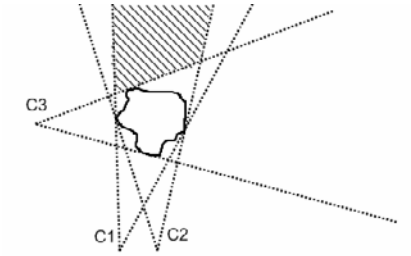


Fig. 2. The first intersection can be between c1 and c3 and not between c1 and c2

Also it'll be easy to determine whether the vertices of P2 fall inside P1. They can be projected onto P1 image plane and a standard algorithm in computational geometry gives the answer.

The edges that connect these vertices are either between existing vertices of P, or between existing vertices of Q, or between an existing point and an intersection point, or between two intersection points.

Tracking the first two kinds of edges is straightforward: if both vertices of an existing edge fall inside the other polyhedron and that edge does not intersect any faces of the other polyhedron, that edge will be retained; if intersections occur, and any of the vertices are inside, it will be connected to the nearest intersection point, and other intersection points will be paired after that; if none of the points are inside, only the intersection points are matched.

Now the extraction of new edges and then faces is no longer a geometrical problem and can be looked at as a graph algorithm.

The two adjacency information sets we have obtained earlier can be used to traverse along the intersection vertices in a graph-like fashion. They are basically unions of some loops (because the original polyhedra were representing solid objects) and these loops are easy to discover with a graph traversal algorithm. For non general position situations, there can be some other structures: single points, single edges, or even "false" loops created by single line. These degenerate cases have to be detected, using information from face normals and the number of faces in each edge.

The extracted edges form polygons that should now be triangulated, again, using Delaunay triangulation to create a mesh representing the intersection.

4 Implementation Results

4.1 The Result of the Recovery on a Synthetic Model of the Popular Utah Teapot Is Presented

One of the original images is shown in **Fig.** . The actual boundary of the teapot is given in **Fig.** -A.

This boundary is Approximated with the progressive schema, subject to various tolerable errors, (**Fig.** -B,C,D) show the result with max error set to 10, 3 and 1 pixels. The numbers of polygon edges are: 18, 35 and 73 respectively.

Fig. 3. Synthetic Utah teapot

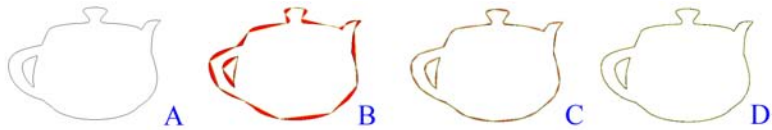


Fig. 4. teapot border (A), the approximate border with for error less than 10 pixels (B), 3 pixels (C) and 1 pixel (D) the difference is painted red for better visibility

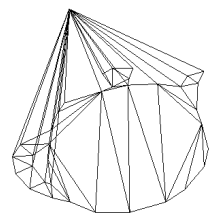


Fig. 5. A single conic area created from one silhouette

Fig. 6. The intersection of all conics for 10 images, from two novel viewpoints

The bounding conic for 3 points tolerance is shown in (**Fig.**). It is composed of 36 vertices and 70 faces. The intersection of all conics for 10 different images is calculated and shown in (**Fig.**). This reconstructed mesh is composed of 848 vertices and 1696 triangular faces. Small artifacts around the handle are caused by random intersections of the conics outside the actual object. These kinds of artifacts are inherent to the nature of shape from silhouette step of the reconstruction and are inevitable, but can be decreased by increasing the number of images used in this step. Other methods of shape from silhouette also demonstrate such effects.

5 Discussion and Justification

The traditional "voxel-based" methods are simpler to implement, but they suffer from several problems:

The first problem is that they have to deal with large number of voxels: a 3D voxel grid with as low as 1000 voxels in each dimension contains 1 billion voxels. Aside from huge amount of memory the whole grid needs, each voxel will require a quite large number of calculations to determine its visibility for each camera. For camera setups in which the object of interest is not completely outside the convex hull of the cameras, this calculation has to be repeated several times.

Another problem with voxel approaches is that they cannot reflect fine measurements of the image, such as sub pixel edge localization [16].

On the other hand a mesh based approach can be set to maintain any level of precision or equivalently, any number of faces. For most common objects, even a fine detailed shape can be represented by a rather small number of faces. The second and major advantage is that this method can use the power of existing common GPUs. The calculation of the intersections can be performed in GPU and program flow and updates be done in the CPU. This will allow a great speed up to the implementation, as well as creating free cycles for CPU to do other tasks. These features are great if a common usage is to be introduced. Finally, there is no need to approximate the shape with a mesh that will be required in all current usages of 3d shape recovery. The resulting polyhedron should just be triangulated using Delaunay triangulation (a fast algorithm is proposed in [17]) or only its polygonal faces be triangulated, see [18].

6 Conclusion

An improved approach to shape from silhouette has been proposed. It uses a polyhedral representation of the visual hull. In order to accelerate the polyhedron to polyhedron intersection problem, the algorithm uses "Intersect" method from DirectX, implemented in hardware. It also generates large non-fragmented faces. Although the implementation is much more complex than voxel based methods, it is more promising to be fast and well suited for subsequent stages, particularly specular analysis.

References

1. Pollefeys, M., Van Gool, L., Vergauwen, M., Verbiest, F., Cornelis, K., Tops, J., Koch, R.: Visual modeling with a hand-held camera. Int. J. Comp. Vision 59(3), 207–232 (2004)
2. Pollefeys, M., Koch, R., Vergauwen, M., Van Gool, L.: Automated reconstruction of 3D scenes from sequences of images. ISPRS J. photogramm remote sens 4(55), 251–267 (2000)
3. Pollefeys, M., Van Gool, L.: Visual modeling: from images to images. J. Vis. Comp. Anim. 13, 199–209 (2002)
4. Kutulakos, K.N., Seitz, S.: A Theory of Shape by Space Carving. Int. J. Comp. Vision, Marr Prize Special Issue 38(3), 199–218 (2000)

5. Yang, R., Pollefeys, M., Welch, G.: Dealing with Textureless Regions and Specular Highlight: A Progressive Space Carving Scheme Using a Novel Photo-consistency Measure. In: International Conference on Computer Vision, pp. 576–584 (2003)
6. Laurentini, A.: The Visual Hull Concept for Silhouette-Based Image Understanding. IEEE Tran. Pattern Anal. Mach. Intell. 16(2) (1994)
7. Eckert, G., Wingbermuhle, J., Niem, W.: Mesh Based Shape Refinement for Reconstructing 3D-Objects from Multiple images. In: 1st European Conference on Visual Media Production (2004)
8. Hosur, P.I., Kai-Kuang, M.: Optimal Algorithm for Progressive Polygon Approximation of Discrete Planar Curves. In: International Conference on Image Processing (1999)
9. Lien, J.M., Amato, N.M.: Approximate Convex Decomposition of Polyhedra. In: ACM Symposium on Solid and Physical Modeling, Beijing, pp. 121–131 (2007)
10. Aronov, B., Sharir, M., Tagansky, B.: The union of convex polyhedra in three dimensions. SIAM J. Comput. 26, 1670–1688 (1997)
11. Tarini, M., Callieri, M., Montani, C., Rocchini, C.: Marching intersections: an efficient approach to shape-from-silhouette. Vis. Model. Visual, 283–290 (2002)
12. Yerex, K.: Real-time Visual Hulls, CMPUT 605 Project Report, April 27 (2004)
13. Matusik, W., Bueler, C., McMillan, L.: Polyhedral visual hulls for real-time rendering. In: 12th Eurographics Workshop on Rendering, pp. 115–125 (2001)
14. Stephenson, M.B., Christiansen, H.N.: A polyhedron clipping and capping algorithm and a display system for three dimensional finite element models. ACM SIGGRAPH Computer Graphics 9(3), 1–16 (Fall 1975)
15. Sutherland, I.P., Hodgman, G.W.: Reentrant Polygon Clipping. CACM 17, 32–42 (1974)
16. Carsten, S.: An Unbiased Detector of Curvilinear Structures. IEEE Tran. Pattern Anal. Mach. Intell. 20(2), 113–125 (1998)
17. Cignoni, P., Montani, C., Scopigno, R.: A fast divide and conquer Delaunay triangulation algorithm in E^d. In: Computer-Aided Design, vol. 30, pp. 333–341. Elsevier, Amsterdam (1998)
18. Chazelle, B.: An Optimal Algorithm for intersecting three-dimensional convex polyhedral. SIAM J. comput. 21(4), 671–696 (1992)
19. Dobrindt, K., Mehlhorn, K., Yvinec, M.: A Complete and Efficient Algorithm for the Intersection of a General and a Convex Polyhedron. In: Workshop on Algorithms and Data Structures (1993)

Discrimination of Bony Structures in Cephalograms for Automatic Landmark Detection

Rahele Kafieh[1], Saeed Sadri[2], Alireza Mehri[1], and Hamid Raji[3]

[1] Department of Biomedical Engineering, Medical University of Isfahan, Isfahan, Iran
r_kafieh@yahoo.com, mehri@cc.mui.ac.ir
[2] Department of Electrical Engineering, Isfahan University of Technology, Isfahan, Iran
sadri@cc.iut.ac.ir
[3] Department of Orthodontics, Faculty of Dentistry Medical University of Isfahan Isfahan, Iran
raji@dnt.mui.ac.ir

Abstract. This paper introduces a new method for automatic landmark detection in cephalometry. In first step, some feature points of bony structures are extracted to model the size, rotation, and translation of skull, we propose two different methods for bony structure discrimination in cephalograms. The first method is using bit slices of a gray level image to create a layered version of the same image and the second method is to make use of a SUSAN edge detector and discriminate the pixels with enough thickness as bony structures. Then a neural network is used to classify images according to their geometrical specifications. Using NN for every new image, the possible coordinates of landmarks are estimated. Then a modified ASM is applied to locate the exact location of landmarks.On average the first method can discriminate feature points of bony structures in 78% of cephalograms and the second method can do it in 94% of them.

Keywords: cephalometry, Active Shape Model, Susan edge detector, Learning Vector Quantization, Bit slicing.

1 Introduction

Cephalometry is a scientific measurement of dimensions of head to predict craniofacial growth, plan treatment and compare different cases. Cephalometry was first introduced by hofrath and Broadbent in 1931, using special holders known as cephalstats to permit assessment of treatment response and of growth. This analysis is based on a set of agreed upon feature points (craniofacial landmarks).

The conventional method of locating landmarks depends on manual tracing of the radiographic images to locate the landmarks. There are ninety landmarks on orthodontics, thirty of which are commonly used by orthodontists [1] (Fig.1, Table.1).

There have been previous attempts to automate cephalometric analysis with the aim of reducing the time required to obtain an analysis, improving the accuracy of landmark identification and reducing the errors due to clinician subjectivity.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 609–620, 2008.

Table 1. List of Landmarks

–	*Abbreviation* –	*Definition* –
1 –	N – –	Nasion, the most anterior point of the nasofrontal in the median – plane
2 –	S –	Sella, The midpoint of the hypophysial fossa –
3 –	A –	Point A, subspindale, the deepest midline point in the curved – bony outline from the base to the alveolar process of the maxilla
4 –	UIT –	Tip of the crown of the most anterior maxillary central incisor –
5 –	UIR –	Root apex of the most anterior maxillary central incisor –
6 –	LIT –	Alveolar rim of the mandible; the lightest, most anterior point on – the alveolar process, in the median plane, between the mandibular central incisors
7 –	LIR –	Root apex the most anterior mandibular central incisor –
8 –	B –	Point B, superamentale, most anterior part of the mandibular – base. It is the most posterior point in the outer contour of the mandibular alveolar process, in the median plane
9 –	Pog –	Pognion, most anterior point of the bony chin, in the median – plane
10 –	Gn –	Gnathion, the most anterior and inferior point of the bony chin –
11 –	Go –	Gonion, a constructed point, the intersection of the lines tangent – to the posterior margin of the ascending ramus and the manibular base
12 –	Me –	Menton, the most caudal point in the outline of the symphysis; it – is regarded as the lowest point of the mandible and corresponds to the anthropological gnathion
13 –	Po –	Porion, most superior point on the head of the condyle –
14 –	Or –	Orbitale, lowermost point of the orbit in the radiograph –
15 –	ANS –	Anterior nasal spine, the tip of the bony anterior nasal spine, in – the median plane –
16 –	PNS –	Posterior nasal spine, the intersection of a continuation of the – anterior wall of the pterygopalatine fosa and the floor of the nose

We can categorize these methods in 4 classes. The first class, being called hand crafted algorithms, usually locates landmarks based on edge detection techniques. In 1986 Levy-Mandel [2] tracked the important edges of image. They used median filter and histogram equalization to enhance the contrast of the image and to remove the noise. Based on their works, in 1989 Parthasarthy [3] presented a similar scheme and reduced the processing time by including a resolution pyramid. Yan [4], Tong [5], Ren [6] and Davis and Taylor [7] presented similar edge tracking methods. Davis and Taylor reported one of the best results of hand crafted algorithms.

All of those edge tracking methods are dependant on the quality of the x-ray and give good results for landmarks on or near to edges in the image.

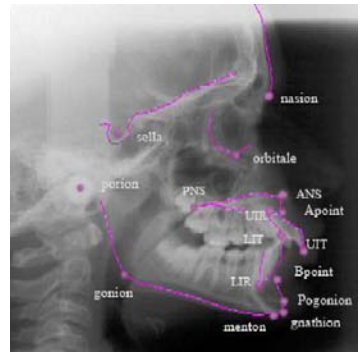


Fig. 1. Location of landmarks

The second class of researchers used mathematical or statistical models to reduce the search area. In 1994 Cardillo and Sid-Ahmed [8] used sub image matching based on gray-scale mathematical morphology. Rudolph (1998) [9] used special spectroscopy to establish the statistical gray model. In 2001Grau [10] improved the work of Cardillo and Sid-Ahmed [8] by using a line detection module to search for the most significant lines, then utilized mathematical morphology approach similar to that used by Cardillo and Sid-Ahmed [8]. In 2000 Hutton [11] used Active Shape Model.

The third class of researchers used Neural Networks, genetic algorithms and fuzzy systems to locate landmarks. In1999 Chen [12] used a combination of Neural Networks and genetic algorithms, without reporting the accuracy of landmark placement. Uchino [13] used a fuzzy machine to learn the relation between the gray levels of image and location of landmarks, but there are many problems with this method. For example they need the same size and rotation and shift for all images, and also a long training time is required. In 2002 Innes [14] used pulse coupled Neural Networks (PCNN) to find regions containing landmarks. In 2004 EI-Feghi et al. [15] used a combination of Neural Networks and fuzzy systems. In 2003 Chakrabartty et al. [16] used support vector machine (SVM) to detect landmarks.

Recently the 4th class of researchers are using a combination of those three classes. In 2006 Yue [17] divided every training shape to 10 regions, and for every region, Principal component Analysis (PCA) is employed to characterize its shape and gray profile statistics. For an input image, some reference landmarks are recognized and the input shape is divided using the landmarks, and then landmarks are located by an active shape model.

2 Material and Methods

2.1 Material

It is important to have a randomly selected data set without any judgment of their quality, sex, age and... . In this research 63 pre-treatment cephalograms were used, which were collected by hutton [11] and were accessible for comparison by other

researchers. The cephalograms were scanned using a Microteck scan Maker 4 flatbed scanner at 100dpi (1pixel = 0.25mm).

A drop-one-out algorithm is used for evaluate results of this method. In this algorithm, each time one of the 63 images is excluded for the test and the algorithm works with the remaining 62 images. So the algorithm can be incorporated 63 times and the mean error is reported as total error.

2.1 Method Overview

Five steps are incorporated in the proposed method. In first step, some features of bony structure are extracted to model the size, rotation and translation of skull. We propose two different methods for bony structure discrimination in cephalograms and compare them to be used in locating some reference landmarks.

The first method is using bit slices of a gray level image to create a layered version of the same image. Firstly, each RGB picture is transformed to a gray leveled one, having two dimensions of width and height. The value of each pixel, playing the role of gray level, can be presented in 8 bits. So a three dimensional structure of width, height and bits can be considered and after a breaking down step, eight black and white versions of bit slices may be created as a new layered image.

In these layers, the ones arising from lower bits of gray level are high frequency pictures and contain fine features of original image, while the ones arising from higher bits of gray level are usually low frequency images and take the overall shape of the picture. Namely in high bit slices, isotropic pixels inside a region are filtered to produce a homogeneous surface and this yields a black and white image of most important parts of skull in highest bit (Fig.2). By decreasing the bit slice level, more features become visible, but noisy structures of high frequency appear and complicate the landmark detection algorithm. So when aiming to detect some reference landmarks on bony structures, it is efficient just to discriminate the overall shape of bony parts in 8^{th} slice of image and trace it to localize three desired landmarks.

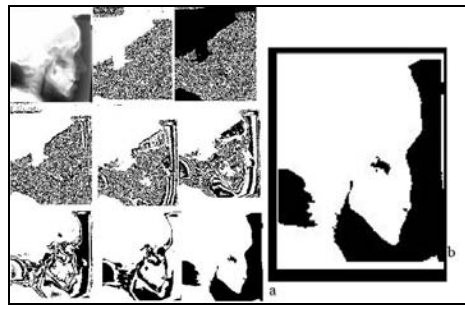


Fig. 2. a) Real Cephalogram and eight bit slices, b) 8^{th} slice

Point Me, is simply located by scanning the image diagonally. The second point, N, is localized in two steps. First the image is scanned horizontally from right to left (to find the first point in frontal bone), and then the image is traced downwards to find the point with the least radius of curvature. The last point, Go, is localized by tracing

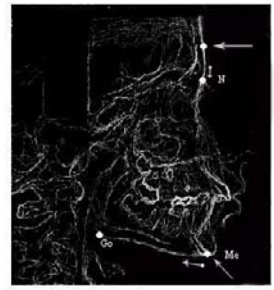


Fig. 3. Automatic detection of points Me, Go and N.

the point Me towards the left ,looking for the least radius of curvature, which is the point Go [17] (Fig 3).

The second method is to make use of a Smallest Univalue Segment Assimilating Nucleus (SUSAN) edge detector [18] and discriminate the pixels with enough thickness as bony structures. For this purpose, firstly a median filter of size 5*5 may be incorporated to reduce the unwanted noisy structures. Then the susan algorithm can be evaluated:

For every pixel of image (x_0, y_0), a circular mask (C(x, y, x_0, y_0)) with a nucleus of (x_0, y_0) is constructed to compare the brightness of nucleus with each pixel inside the mask (Fig.4).

$$C(x,y;x_0,y_0) = \begin{cases} 1 & if \quad |I(x,y) - I(x_0,y_0)| \le t \\ 0 & else \end{cases} \tag{1}$$

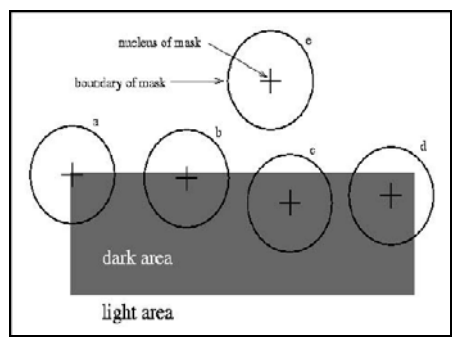


Fig. 4. Circular mask of susan edge detector

It means that amount of ones in disk increases when more pixels in disk have similar brightness with nucleus, so for smooth areas, C has more ones and in edges, C has more zeros. Then a value of n is defined as the area of ones in disk:

$$n(x_0, y_0) = \sum_x \sum_y C(x, y; x_0, y_0) \tag{2}$$

And the edge response is defined:

$$R(x_0, y_0) = \begin{cases} \frac{3}{4} n_{\max} - n & if \quad n \prec n_{\max} \\ 0 & else \end{cases} \tag{3}$$

Values of t and n_{max} may be selected according to image complexity and desired resolution. Now a new image of R values represents the edges of original image. There are many advantages in using susan edge detection method, for example it is 10 times faster than canny, gives good edge connectivity and good tolerance to noise. Also without incorporating edge thinning, thickness of edges is proportional to edge intensity, so selecting the thick edges is the same as selecting intense ones, arising from bony structures (Fig.5).

To evaluate the thick ness, edge direction is calculated and the connected pixels in that direction, with value more than a predefined threshold are counted. For this purpose, the longest axis of symmetry is defined as edge direction. So the moments are taken and ratio of second moments in x and y determines the orientation, and sign of xy moment shows the positive or negative gradient:

$$\begin{aligned} M_x &= \sum_x \sum_y (x - x_0)^2 C(x, y; x_0, y_0) \\ M_y &= \sum_x \sum_y (y - y_0)^2 C(x, y; x_0, y_0) \\ M_{xy} &= \sum_x \sum_y (y - y_0)(x - x_0)\ C(x, y; x_0, y_0) \\ sign &= \begin{cases} + & if \quad M_{xy} \geq 0 \\ - & if \quad M_{xy} \prec 0 \end{cases} \\ \theta &= sign \cdot Arctg(\frac{M_y}{M_x}) \end{aligned} \tag{4}$$

In order to determine a proper threshold to select the thick parts, the mean value of θ over the whole shape is calculated and threshold is set to be $2/3\,\theta$. It is important to have an adaptive threshold, because Cephalogram images of this study may have different brightness or be taken with different apparatus, so selecting a fix threshold doesn't give enough satisfaction.

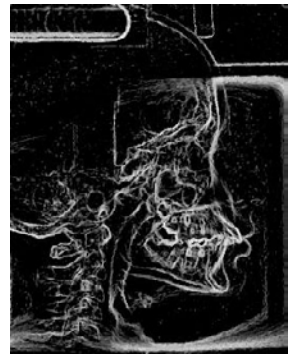


Fig. 5. Susan edge detection

Finally when aiming to detect some reference landmarks on bony structures, it is efficient just to discriminate the important edges of bony parts and trace it (in the way described for first method) to localize three desired landmarks (Fig.3).

After detecting the three important points, some measurement features are calculated using their coordinates. These features are described in Fig.6.

The second step of algorithm is applying a data clustering on measurement features, to classify cephalograms according to their geometrical specifications. A k-means clustering was used for this dataset and the algorithm converged to 5 distinct clusters. Inputs of this clustering system are features of Fig.4, which were normalized with GoN as the unit length.

In the third step, we propose that results of k-means clustering be used as a target for training the LVQ Neural Networks.

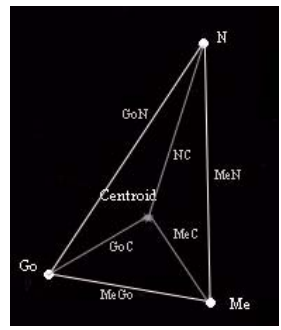


Fig. 6. Calculating some measurement features

From now on, every test image can be evaluated with trained LVQ network to be classified in one of those 5 clusters.

The fourth step of this method is estimating the possible coordinates of landmarks for every new image. In order to do this, knowing the correct cluster of each new image (in step 3), all of the training images of that cluster were aligned to the new image and for every landmark the average of coordinates on aligned training images

Fig. 7. Estimation of coordinates

were calculated. Alignment procedure in this step can be done having coordinates of 3 important points and with a linear conformal alignment (Fig.7).

The fifth step of algorithm is a modified Active Shape Model (ASM). The key difference between this method and a conventional ASM [11, 19] is the start shape. In a conventional ASM the start shape is the same as mean shape calculated from the training set. While in this modified ASM the start shape passes through some closer points to the correct landmarks position and increases the probability of convergence to correct position. So the mean shape changes in a way that passes through the estimated coordinates of landmarks (in step 4). In order to change the mean shape to fit the estimated landmarks, the complete annotated structure should be broken to some smaller structures, which can be considered to be independent from each other and each of them passes through (at least) 3 of estimated landmarks. Now there are 3 points in each sub image, so that affine alignment which consists of 3 equations can be used on them (Fig.8).

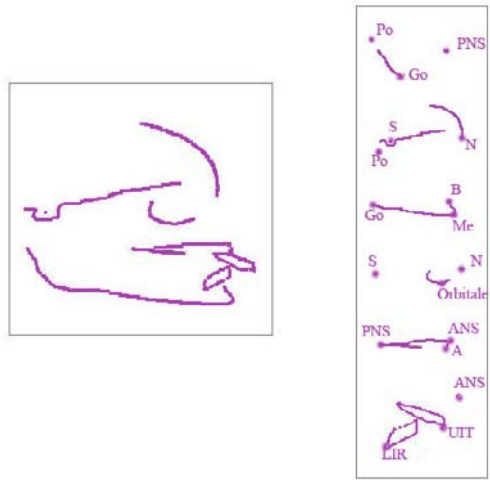


Fig. 8. The complete annotated structure broken to some smaller structures

The training set of input images which are annotated by the expert are used, firstly to calculate their mean shape with a Procrusters algorithm [17] and secondly PCA (Principal Component Analysis) was incorporated to find the main eigenvectors of the set and characterize its shape and gray profile statistics.

Now to find the best match to the intensity profile, start shape is overlaid on the image and the local search is performed to move each point to get the best location (Fig.9). In this paper, a multi resolution approach [20] was used to reduce the errors and get the better fit.

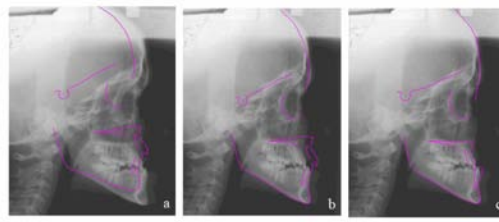


Fig. 9. a)The mean shape, overlaid on image, b)The mean shape after alignment with estimated coordinates, c)After convergence of ASM algorithm

3 Results

As it is illustrated in Fig.2 and Fig.4, the results of first step which is a novel method on cephalograms shows a distinct improvement on other similar works, as the result of bony structure discrimination to reduce errors of wrong landmark detection on soft tissue. It can be seen that the second method gives better response than the first one. Because in some images of high brightness, bit slicing discriminates over brighten soft tissue wrongly, but adaptive thresholding in second method, prevents such a problem and discriminate bony structures, as correctly as possible (Fig.10).

An LVQ neural network is proposed to be used for classification in our algorithm. The most important advantage of this method is in cases where we are in lack of

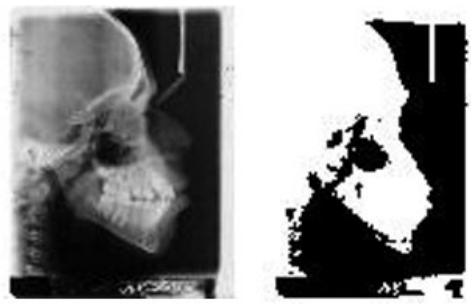


Fig. 10. Bit slicing discriminates over brighten soft tissue wrongly

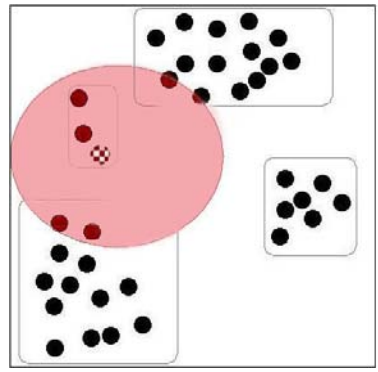


Fig. 11. Patterned circle: new test object, Ellipse: wrong classification, Rounded rectangle: correct classification

enough big training set. Yue [17] proposed a method to measure the similarity between new image and all the training images (a big enough set of around 250 images), by Euclidian distance between their feature vectors (Fig.5). For an input image, the top 5% most similar images in training set were selected to be aligned and averaged to estimate the landmark locations.

But where datasets are small (like the one in this paper), it is possible to have just 1 or 2 really similar images to the new image and all of 5% of images (for example 4 images), are not really similar which makes the estimated results unacceptable (Fig.11). Results of estimation in this method shows if we stop the algorithm in this step, the mean error will be about 3.4 mm.

If the algorithm stops in fifth step, the mean error of landmark locating with modified ASM will be about 2.8 mm. Comparing to a conventional ASM [11], the error rate is improved about 5.5mm. (The mean error rate was 8.3 mm for conventional method.)

4 Discussion

This paper introduces a new method for automatic landmark detection in cephalometry. There are many parameters in this method, which should be studied and optimized later. Among which are t and n_{max} parameters of susan edge detector.

Also many other methods can be evaluated, like fuzzy clustering, using template matching during ASM, and classifying the images to different classes of gray profile of whole image and finally using other model based approaches like AAM and medial profiles.

5 Conclusion

In order to evaluate the results of this method, 63 randomly selected images were used with a drop-one-out method.

On average the first method can discriminate feature points of bony structures in 78% of cephalograms and the second method can do it in 94% of them. As a final result 24 percent of the 16 landmarks are within 1 mm of correct coordinates, 63 percent within 2 mm, and 95 percent within 5 mm, which shows a distinct improvement on other proposed methods as described in table.2.

Table 2. Comparison of results

Methods	Recognition Rate		
	Within 1mm	Within 2mm	Within 5mm
Hutton's method	13%	35%	75%
Yue's method	Not reported	57%	Around 88%
Our method	24%	61%	93%

Acknowledgments. The authors would like to appreciate Dr Ramin Kafieh, DDS, for his useful guides in dental problems, and also acknowledge the Eastman Dental Institute as the source of the cephalogram images.

References

1. Rakosi, T.: An Atlas and Manual of Cephalometric Radiology. Wolfe Medical, London (1982)
2. Luevy-Mandel, M., Venetsanopoulos, A., Tsotsos, J.: Knowledge based landmarking of cephalograms. Comput. Biomed. 19, 282–309 (1986)
3. Parthasaraty, S., Nugent, S., Gregson, P.G., Fay, D.F.: Automatic landmarking of cephalograms. Comput. Biomed. 22, 248–269 (1989)
4. Yan, C.K., Venetsanopoulos, A., Filleray, E.: An expert system for landmarking of cephalograms. In: Proceedings of the Sixth International Workshop on Expert Systems and Applications, pp. 337–356 (1986)
5. Tong, W., Nugent, S., Gregson, P., Jesen, G., Fay, D.: Landmarking of cephalograms using a microcomputer system. Comput. Biomed. 23, 358–397 (1990)
6. Ren, J., Liu, D., Shao, J.: A knowledge-based automatic cephalometric analysis method. In: Proc. 20th Annu. Int. Conf. IEEE Engineering in medicine and biology Soc., pp. 123–127 (1998)
7. Savis, D., Taylor, C.: A blackboard architecture for automating cephalometric analysis. Med. Inf. 16, 137–149 (1991)
8. Cardillo, J., Sid-Ahmed, M.: An image processing system for locating craniofacial landmarks. IEEE Trans. Med. Imag. 13, 275–289 (1994)
9. Rudolph, D., Sinclair, P., Coggins, J.: Automatic computerized radiographic identification of cephalometric landmarks. Am. J. Orthod. Dentofec. Orthop 113, 173–179 (1998)
10. Grau, V., Juan, M., Monserrat, C., Knoll, C.: Automatic localization of cephalometric landmarks. J. Biomed. Inf. 34, 146–156 (2001)
11. Hutton, T.J., Cunningham, S., Hamrnond, P.: An evaluation of active shape models for the automatic identification of cephalometric landmarks. European Journal Orthodontics 22, 499–508 (2000)

12. Chen, Y., Cheng, K., Liu, J.: Improving cephalogram analysis through feature subimage extraction. IEEE Eng. Med. Biol. Mag. 18, 25–31 (1999)
13. Uchino, E., Yamakawa, T.: High speed fuzzy learning machine with guarantee of global minimum and its application to chaotic system identification and medical image processing. In: Proceeding of Seventh International Conference on tools with Artificial Intelligence, pp. 242–249 (1995)
14. Innes, A., Ciesielski, V., Mamutil, J., John, S.: Landmark detection for cephalometric radiology images using pulse coupled neural networks. In: Int. Conf. in computing in communication, pp. 391–396 (2002)
15. El-Feghi, I., Huang, S., Sid-Ahmed, M.A., Ahmadi, M.: X-ray Image Segmentation using Auto Adaptive Fuzzy Index Measure. In: The 47th IEEE International Midwest Symposium on Circuits and Systems, pp. 499–502 (2000)
16. Chakrabartty, S., Yagi, M., Shibata, T., Gawenberghs, G.: Robust cephalometric landmark identification using support vector machines. In: Proc. Int. Conf. Multinedia and Expo., pp. 429–432 (2003)
17. Yue, W., Yin, D., Li, C.H., Wang, G.: Locating Large-Scale Craniofacial Feature Points on X-ray Images for Automated Cephalometric Analysis. IEEE, Los Alamitos (2005)
18. Smith, S.M.: (n.d.) [Online], SUSAN Low Level Image Processing (2007), http://users.fmrib.ox.ac.uk/~steve/susan
19. Cootes, T.F., Taylor, C.J.: Active shape models. In: Hogg, D., Boyle, R. (eds.) 3rd British Machine Vision Conference, pp. 266–275 (1992)
20. Cootes, T.F., Taylor, C.J., Lanitis, A.: Active shape models: Evaluation of a multi-resolution method for improving image search. In: Hancock, E. (ed.) 5th British Machine Vison Conference, pp. 327–336 (1994)

Grid Based Registration of Diffusion Tensor Images Using Least Square Support Vector Machines

Esmaeil Davoodi-Bojd[1] and Hamid Soltanian-Zadeh[1,2]

[1] Control and Intelligent Processing Center of Excellence, School of Electrical and Computer Engineering, University of Tehran, Tehran 14395-515, Iran
es.davoodi@ece.ut.ac.ir, hszadeh@ut.ac.ir
[2] Image Analysis Laboratory, Radiology Department, Henry Ford Hospital, Detroit, MI 48202, USA
hamids@rad.hfh.edu

Abstract. In this paper, we present a non-rigid image registration method for DTMR images. This method consists of finding control points using a piecewise affine registration procedure and then estimating final transform between two images by minimizing corresponding Least Squares Support Vector Machine (LS-SVM) function of these control points. In our scheme, a fully symmetric grid points in the reference image is selected and the transformed grid points are computed using the results of piecewise affine registration. These control points are then employed to estimate final transform between images by minimizing the related LS-SVM function. In the transform functions, a finite strain (FS) based reorientation strategy is applied to adopt these methods for DTMR images. The main advantage of this method is that in estimating transform function, it considers all control points. Thus, each point in the reference image is transformed consistently with all other image points.

Keywords: Diffusion Tensor MRI, Image Registration, Piecewise Affine Transform, Tensor Reorientation, Least Square Support Vector Machines.

1 Introduction

Diffusion Tensor Magnetic Resonance Imaging (DTMRI) is a noninvasive tool for determining white matter connectivity in the brain. DTMRI adds to conventional MRI the capability of measuring the random motion of water molecules, referred to as diffusion. It has been known that the water molecule motion is restricted in the axons due to the existence of myelin sheath [1, 2]. Therefore, the most important distinctive characteristic of DTMR images is that they have directional information of microtubule living structures. Direction at each voxel is computed mathematically using a 3×3 symmetric positive semi-definite matrix $\boldsymbol{D}$, known as Diffusion Tensor (DT), provided by DTMRI. Consequently, working with and processing of DTMR images are more complicated than conventional image modalities.

Image registration is required for group normalization, atlas construction, and automatic multi structure segmentation. However, applying conventional image registration methods to the DTMR images without considering orientation information of these images does not yield acceptable results. Therefore, it has been proposed to use

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 621–628, 2008.

tensor reorientation strategies when applying spatial transformation to DTMR images [3]. One of these strategies is called Finite Strain (FS) method which is sufficient for common image registration and normalization procedures. FS decomposes the transform function into a rotation matrix and a deformation matrix, and then applies the rotation matrix to the tensors such that directions of eigenvectors of each tensor rotate consistently with the original spatial transform.

In [4], different combinations of channels for image registration were used, including four channels of scalar characteristics of diffusion tensor and one channel of components of diffusion tensor. Although they did not use reorientation strategy in the last channel, but they concluded that using whole diffusion tensor components in image registration generates better results than other combinations.

Several non-rigid DTMR image registration procedures have also been proposed. Some of them use affine registration as a basis and then by applying it in different parts of the images find a non-linear transform to match the images [5, 6]. On the other hand, some methods use multi-resolution schemes by increasing the complexity of transform function in each level of registration [7]. One of the important issues of piecewise affine registration is the problem of combining resulting transforms which belong to different parts of images. In [5], an interpolation method between neighbored transforms was proposed. Unfortunately, this solution only considers limited number of neighbors to estimate transformed border voxels. Thus, the resulting transformed regions may be inconsistent with the whole transformed image.

In [8], a scheme for image registration using least square support vector machines (LS-SVM) was presented. The important capability of this method is that it estimates a non-linear transform between two images using some control points from the images. However, this method has not been applied to DTMR images.

The aim of is this work is to develop a non-rigid image registration method for DTMR images using piecewise affine transform and least square support vector machines. In our scheme, we first compute some control points from the two images. These control points are used to estimate a transform function between the two images by minimizing their LS-SVM function, in combination with reorienting the diffusion tensors during the transform.

2 Methods

2.1 Piecewise Affine Registration

An affine transform, $A_p(.)$, can be expressed by 12 parameters: 3 parameters for rotation ($\underline{q}$), 6 for deformation ($\underline{s}$), and 3 for translation ($\underline{t}$). Transformed point, $\underline{y}$, of a point, $\underline{x}$, can be computed by (1), in which $\boldsymbol{Q}$ is the 3×3 rotation matrix with 3 independent parameters, $\boldsymbol{S}$ is the 3×3 deformation matrix with 6 independent parameters, T is the 3×1 translation vector, and finally, $\underline{p}$ is the whole unknown parameter vector consists of [$\underline{q}$, $\underline{s}$, $\underline{t}$].

$$\underline{y} = (Q.S).\underline{x} + T = A_{\underline{p}}(\underline{x}). \tag{1}$$

Image registration can be formulated as a minimization problem of a dissimilarity criterion between two images. However, for DTMR images this criterion must handle

the orientation of the tensors. Reference [5] has used a dissimilarity function with FS reorientation strategy for affine registration, as shown in (2).

$$\phi(\underline{p}) = \int_{\Omega} \left\| I_s((Q.S).\underline{x} + T) - Q.I_r(\underline{x}).Q^t \right\|^2 d\underline{x}. \tag{2}$$

==========

Begin

- *Divide the two DT images into $n_x \times n_y \times n_z$ equal size sub-images.*
- *For each corresponding pair of sub-images (Ω^s_{ijk} and Ω^r_{ijk}, i=1,…, n_x, j=1,…, n_y, k=1,…, n_z) do the following steps:*
  1. *Double each dimension of each sub-image by adding zero voxels to their peripheral sides. (This is essential in order to let transformed image rotate, translate, or scale beyond sub-image size).*
  2. *Apply evolutionary algorithm with cost function given in (3) to find an initial estimation $\underline{p}^0_{ijk}$ of unknown parameters.*
  3. *Apply CG algorithm using initial point, $\underline{p}^0_{ijk}$, to minimize dissimilarity function given in (2) and find affine transform parameters $\underline{p}_{ijk}$.*

end.

==========

Fig. 1. Piecewise affine registration algorithm of DT images

where $\boldsymbol{I_s(.)}$ and $\boldsymbol{I_r(.)}$ are sensed and reference images, respectively, and Ω is the region in which the dissimilarity measure is computed. The second term of integral, $Q.\boldsymbol{I_r}(\underline{x}).Q^t$, reorients the diffusion tensor in each voxel. This function can be minimized by gradient based optimization methods because its gradient expressions can be computed. In [5], the gradients of this function have been computed which we use in a Conjugate Gradient (CG) algorithm to find unknown transform parameters.

One of the basic problems of gradient based optimization methods is that they get easily trapped in local minima. Therefore, we use an Evolutionary Algorithm (EA) to find the general region of the optimal parameters before applying the CG algorithm. In EA, the unknown parameters ($\underline{p}$) may have any values. This may lead the algorithm to too far solutions if the transformed point, i.e., $\underline{y}$, lays out of range of the images (for example $\underline{y}=[-10,-20,3]^t$). Therefore, the cost function for EA should consider these out-of-the-range-points. Our proposed cost function for EA is:

$$\psi(\underline{p}) = \phi(\underline{p}) + \frac{\sum_{\underline{x} \in \Omega} isOut(A_{\underline{p}}(\underline{x}))}{N_{\Omega}}, \qquad where: \begin{cases} isOut(\underline{y}) = 1, \underline{y} \notin \Omega. \\ isOut(\underline{y}) = 0, \underline{y} \in \Omega. \end{cases} \tag{3}$$

where N_{Ω} is the total number of points in Ω. The summary of our piecewise affine registration algorithm is presented in Fig. 1. The remaining problem is: how to combine the resulting sub-transforms to build final transformed image. This problem arises mainly in the border voxels of sub-images. Reference [5] solves this problem by finding new sub-transforms for those voxels using interpolation between neighboring sub-transforms. However, this solution only considers limited number of neighbors to estimate the transformed border voxels. We handle this problem by con-

sidering a grid of points in the reference image (we will discuss about the strategy of selecting these control points in section 3.2), and calculate the transformed points using the corresponding sub-transforms. Then, we employ these points as control points to estimate transformed image using LS-SVM.

2.2 Transformation Estimation Via LS-SVM

Suppose we have two corresponding point sets $\boldsymbol{S}$ and $\boldsymbol{R}$ belonging to the two images, respectively, and each set consists of N spatial control points. The aim of transformation estimation is to find a transform which maps the sensed image control points, i.e., $\boldsymbol{S}$, to the corresponding reference image control points, i.e., $\boldsymbol{R}$. We use a model with linear combination of Radial Basis Functions (RBF) as proposed in [8]:

$$\underline{y} = f(\underline{x}) = \sum_{i=1}^{N} \underline{a}_i . \exp\left\{ -\left\| \underline{x} - \underline{x}_i \right\|^2 / \sigma^2 \right\} + \underline{b}. \tag{4}$$

where $\underline{x}$ is the coordinate of a spatial point in the sensed image and $\underline{y}$ is the estimated corresponding point in the reference image, and $\{\underline{x}_i, i=1,\ldots,N\}$ are N control points in the sensed image. The coefficients $\underline{a}_i$ and $\underline{b}$ are computed using (5) [8], in which $\boldsymbol{Y}$ is a $3\times N$ matrix representing N control points in the reference image, $\mathbf{1}$ is a $1\times N$ vector consists of N ones, and $\boldsymbol{\Omega}$ is a $N\times N$ matrix computed by (6).

$$\underline{a} = (Y - \underline{b}.1).\Omega^{-1} \qquad , \qquad \underline{b} = \frac{Y.\Omega^{-1}.1^T}{1.\Omega^{-1}.1^T}. \tag{5}$$

$$\begin{cases} \Omega_{ij} = \exp\{-\left\| x_i - x_j \right\|^2 / \sigma^2\}, & i \neq j. \\ \Omega_{ij} = \exp\{-\left\| x_i - x_j \right\|^2 / \sigma^2\} + \gamma^{-1}, & i = j. \end{cases} \tag{6}$$

The parameters σ and γ are tuning parameters. While σ controls the depth of contribution of each control points in its neighborhood, γ determines a tradeoff between the model complexity and training error [8]. Reference [8] proposed an adaptive scheme to determine these parameters. But, usually they can be chosen by prior knowledge or empirically.

2.3 DT Reorientation for the Estimated Function

Equation (4) only estimates the transformation function between two images. However, for DTMR images a reorientation strategy must be applied. FS reorientation strategy needs a rotation matrix Q for each tensor. For affine transforms, this matrix can be computed easily, but for this nonlinear function more calculations must be performed. Suppose for each voxel, the nonlinear function in (4) can be approximated by an affine transform (7), in which $F_{\underline{x}}=Q_{\underline{x}}.\ S_{\underline{x}}$ is the linear transformation matrix.

$$\underline{y} = f(\underline{x}) \equiv F_{\underline{x}}.\underline{x} + T. \tag{7}$$

$$J_f(\underline{x}) = -\frac{2}{\sigma^2} \times \sum_{i=1}^{N} \underline{a}_i.(\underline{x} - \underline{x}_i).\exp\left\{-\left\|\underline{x} - \underline{x}_i\right\|^2 / \sigma^2\right\} \equiv F_{\underline{x}}. \tag{8}$$

Differentiating both sides of (7) with respect to $\underline{x}$ gives (8), in which $J_f(\underline{x})$ is the Jacobian matrix of $f(.)$. Now the rotation matrix can be estimated from the computed $F_{\underline{x}}$ by (9) [3]. Finally, the transformed image $I_t(.)$ of the reference image $I_r(.)$ can be expressed by (10) which consists of both spatial and tensor reorientation transforms.

$$Q_{\underline{x}} = (F_{\underline{x}}.F_{\underline{x}}^t)^{-0.5}.F_{\underline{x}}. \tag{9}$$

$$I_t(f(\underline{x})) = Q_{\underline{x}}.I_r(\underline{x}).Q_{\underline{x}}^t. \tag{10}$$

3 Results and Discussion

3.1 Pre-processing

In this work, we used two DTMRI data sets acquired at Henry Ford Hospital using a 1.5T GE MRI (General Electric medical systems, Milwaunkee, WI, USA) from two healthy volunteers. The voxel size was 0.9375×0.9375×3 mm and the image size was 256×256×40. For each slice, 6 diffusion-weighted images and one T2-weighted image were acquired. In order to reduce the computation time, we downsampled each image slice to 128×128. This does not affect the generality of the problem.

For each slice, we extracted the brain using the corresponding 6 diffusion weighted-images and used this as a mask to remove the background. For building this mask, we averaged the 6 diffusion-weighted images and applied a simple thresholding process to the resulted image. Two samples of this mask are shown in Fig. 2 (a, b). Then, we calculated the tensors using the method of [9]. In Fig. 2 (c, d) the first diffusion tensor component for the reference and subject images are illustrated.

After calculating diffusion tensors, the resulting images were aligned coarsely using the affine registration algorithm in Fig. 1 with $n_x=n_y=n_z=1$. The result of the coarse matching is illustrated in Fig. 2 (e). Note that in this work, we applied each algorithm on the whole 3D diffusion tensor images and only first component of diffusion tensor of a typical slice is shown in Fig. 2.

3.2 Main Registration Procedure

The main registration process consists of finding control points using piecewise affine registration and then estimating the final transform between the two images by minimizing LS-SVM functin of these control points.

In the piecewise affine registration algorithm, Fig. 1, we chose $n_x=n_y=16$ *and* $n_z=5$. Therefore, by extracting 8 control points from each sub-image, 16×16×5×8=10240

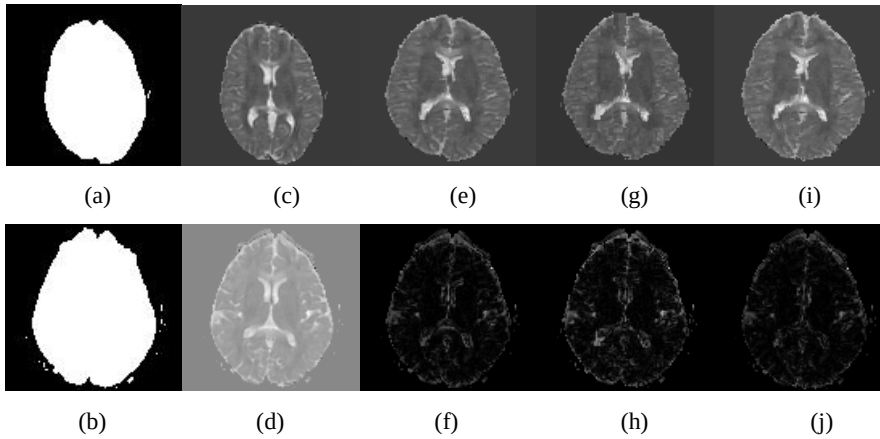


Fig. 2. (a) - (b), computed masks for eliminating background noise, corresponding to reference and subject images, respectively. (c) - (d), first component of diffusion tensor of reference and subject images, respectively. (e), result of coarse matching of the images. (f), absolute difference between the subject image and (e). (g), result of fine matching using the method of [5]. (f), absolute difference between the subject image and (g). (i), result of fine matching using our proposed method. (j), absolute difference between the subject image and (i). Note that the proposed registration method improved the matching.

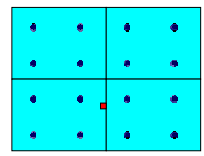


Fig. 3. Illustration of a fully symmetric grid. Blue circles are the control points and the red square is a border point that we want to estimate its transformed point from the control points.

Table 1. Normalized sum of square error between the images for each matching method: coarse matching, fine matching using the method of [5] (fine matching 1) and fine matching using our proposed method (fine matching 2). The errors were computed for the 6 independent components of diffusion tensors in entire image data set.

Tensor Component	**Matching error (%)**			
	before matching	coarse matching	fine matching 1	fine matching 2
D_{xx}	15.42	12.89	11.79	10.82
D_{xy}	09.68	07.65	06.60	06.09
D_{xz}	12.79	11.73	11.70	11.08
D_{yy}	09.95	08.06	07.32	06.89
D_{yz}	10.79	10.03	09.65	09.01
D_{zz}	13.68	10.13	09.68	08.94
Whole Tensor	12.05	10.08	09.46	08.81

pairs of control points were computed. It may appear that by increasing the number of sub-images and consequently by increasing the number of control points, the resulting transform function is more precise, but the size of sub-image should be large enough so that its voxels have enough information for registration process.

Another important issue about control points is that how some points from each sub-image are selected as control points. As we used piecewise affine registration to match the images, two key points should be considered:

- Since for each pair of sub-images an affine transform was computed, at least four non-planar spatial points should be selected from each sub-image.
- The control points should be located symmetrically in the reference sub-image space (see Fig. 3). This consideration is the result of the fact that in the LS-SVM method, in order to estimate transformed point of a point in the reference image, the distances between that point and all control points are computed (see (4)). Therefore, the points on the borders (for example the red square in Fig. 3) should have equal distances from symmetric control points on the neighbored sub-images so that they get equal effects from neighboring transforms.

Consequently, we selected 8 corner points of a cube with half size in each dimension, centered on each sub-image to build the grid in the reference image.

After calculating control points grid, we estimated final transformed image, which is shown in Fig. 2 (i), using the method introduced in section 2.2. For better comparison, we also applied the method of [5] (Fig. 2 (g)). In this figure, the absolute difference between the subject image and final transformed images is also shown. By comparison the images in Fig. 2 (g, h, i, j), it can be seen that the resulted image of our method is more similar to the subject image and there is no inconsistent regions thanks to the capability of LS-SVM method to estimate a smooth transform function.

To quantify the similarity measure between images, we calculated the normalized sum of square error between the images for each method as shown in Table 1. For better comparison the matching error of each component of diffusion tensor was also computed. For computing the error, we normalized sum of square difference between the images with respect to the total number of voxels and the variation range of values of the images. By considering this fact that the matching process has been applied on the whole tensor, it can be seen from this table that the matching results are good for each tensor component as well as the whole tensor. The last column of this table also shows quantitatively the preference of our method compared to the method of [5].

4 Conclusion

In this paper, we presented a non-rigid image registration method for DTMR images. This method consists of finding control points using a piecewise affine registration and then estimating final transform between two images by minimizing LS-SVM function of these control points. In the piecewise affine registration, the images are divided into equal size sub-images. Then, an evolutionary algorithm is applied to find an initial solution for the affine transform parameters in each pair of sub-images. After that, a CG algorithm is applied to refine this solution. Finally, a fully symmetric grid of points in the reference image is selected and the transformed grid is computed

using the results of piecewise affine registration. These control points are then employed to estimate the final transform between the images by minimizing the related LS-SVM function. In both piecewise affine registration and final transform estimation, an FS based reorientation strategy is applied.

One of the advantages of this method is that in estimating the transform function using control points, it considers all control points and thus all transforms of sub-images. Therefore, each point in the reference image is transformed consistently with the entire image. Another advantage of this procedure is that it can be applied with different sub-image sizes and consequently with different grid sizes. This can help us to estimate fine transforms between images from coarse to fine resolutions.

References

1. Goldberg-Zimring, D., Mewes, A.U., Maddah, M., Warfield, S.K.: Diffusion tensor magnetic resonance imaging in multiple sclerosis. American Society of Neuroimaging 15, 68S–81S (2005)
2. Westin, C.F., Maier, S.E., Mamata, H., Nabavi, A., Jolesz, F.A., Kikinis, R.: Processing and visualization for diffusion tensor MRI. Med. Imag. Analysis 6, 93–108 (2002)
3. Alexander, D.C., Pierpaoli, C., Basser, P.J., Gee, J.C.: Spatial transformations of diffusion tensor magnetic resonance images. IEEE Trans. Med. Imag. 20(11), 1131–1139 (2001)
4. Park, H.J., Kubicki, M., Shenton, M.E., Guimond, A., McCarley, R.W., Maier, S.E., Kikinis, R., Jolesz, F.A., Westin, C.F.: Spatial normalization of diffusion tensor MRI using multiple channels. Neuroimage 20(4), 1995–2009 (2003)
5. Zhang, H., Yushkevich, P.A., Alexander, D.C., Gee, J.C.: Deformable registration of diffusion tensor MR images with explicit orientation optimization. Med. Imag. Analysis 10, 764–785 (2006)
6. Zhang, H., Yushkevich, P.A., Gee, J.C.: Registration of diffusion tensor images. In: CVPR 2004, vol. 1, pp. 842–847 (2004)
7. Cao, Y., Miller, M.I., Mori, S., Winslow, R.L., Younes, L.: Diffeomorphic matching of diffusion tensor images. In: CVPRW 2006, pp. 67–74 (2006)
8. Peng, D.Q., Liu, J., Tian, J.W., Zheng, S.: Transformation model estimation of image registration via least square support vector machines. Pattern Recognition Letters 27(12), 1397–1404 (2006)
9. Besser, P.J., Jones, D.K.: Diffusion-tensor MRI: theory, experimental design and data analysis –a technical review. NMR in Biomedicine 15, 456–467 (2002)

Detection of Outer Layer of the Vessel Wall and Characterization of Calcified Plaques in IVUS Images

Alireza Roodaki[1], Zahra Najafi[1], Armin Soltanzadi[1], Arash Taki[2], Seyed Kamaledin Setarehdan[1], and Nasir Navab[2]

[1] Control and Intelligent Processing Centre Of Excellence , Faculty of Electrical and Computer Engineering, University of Tehran, Tehran, Iran
{a.roodaki@ece.ut.ac.ir,ksetareh@ut.ac.ir}
[2] Computer Aided Medical Procedures (CAMP) - TU Munich, Germany

Abstract. Intravascular Ultrasound (IVUS) is a diagnostic imaging technique that provides tomographic visualization of coronary arteries. Important challenges in analysis of IVUS images are speckle noise, artifacts of catheter and calcified shadows. In this paper, we present a method for the automated detection of outer (media-adventitia) border of vessel by the use of geometric deformable models. Speckle noise is reduced with median filter. The initial contour is extracted using Canny edge detection and finally the calcified regions are characterized by using Bayes classifier and thresholding methods. The proposed methods were evaluated on 60 IVUS images from 7 different patients. The results show that the border detection method was statistically accurate and in the range of inter observer variability (based on the used validation methods). Bayesian classifier enables us to characterize the regions of interest, with a sensitivity and specificity of 92.67% and 98.5% respectively.

Keywords: Border detection, Deformable models, IVUS, Tissue characterization.

1 Introduction

Intravascular Ultrasound (IVUS) is a catheter-based medical imaging technique. Using a specially designed ultrasound catheter it provides real-time tomographic images of the arterial wall that shows the morphology and histological properties of the cross-section of the vessel. IVUS not only provides a quantitative assessment of the vessels' wall but also introduces information about the nature of atherosclerotic lesions as well as the plaque shape and size [1]. The first step for plaque characterization in IVUS images is detection of outer layer of vessel wall. Nevertheless, it is a difficult, subjective and time-consuming procedure to manually perform segmentation. Therefore, there is an increasing interest in developing automatic tissue segmentation algorithms for IVUS images.

Reducing the effect of the speckle noise is required for many applications in ultrasound image processing [2]. Most widely used techniques to reduce the speckle noise include the Median Filter. This filter seeks a balance between averaging and all pass filters, although it improves the quality of the images. Several segmentation methods

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 629–636, 2008.

in IVUS images have been proposed in the literature. Some of the earlier works on segmentation of the IVUS images were semi-automatic and some other were based on the energy minimization of a contour either guided snake toward the target structure or minimize a cost function [5, 6]. Segmentation methods based on probabilistic approaches have been proposed in [4, 8]. Some authors combine transversal and longitudinal contours to provide the model with spatial continuity along the sequences [7]. Region growing method is used in [9] to detect the edges.

The active contours used in the previous approaches are mostly based on a kind of parametric deformable model. However, parametric deformable models have two main limitations [10]. For recognizing calcium regions in IVUS images some characterization methods have been proposed using adaptive thresholding algorithm [3] and texture based features [13, 14]. However the thresholding based method is sharp in discriminating and ignores the variants of gray level of regions, and the latter is time consuming and doesn't have the desirable accuracy.

We have focused on the development and validation of an automated method based on non-parametric deformable models for accurate IVUS image segmentation. The results of the application of the proposed algorithm to real IVUS images of different patients are presented. After detecting the media-adventitia boundary and extracting the region between this border and the catheter, the Bayes classifier and the thresholding based method is employed to characterize the calcified regions. The calcified area detected by the implemented methods was validated and compared with the manually segmented areas by the expert.

2 Method

2.1 Border Detection

Reducing the effect of the speckle noise is required for many applications in ultrasound image processing. Median filter is used for denoising in this work [2]. The planar image in Cartesian coordinates is converted into the polar coordinates. The reason for this is that the more or less circular vessel structures can be processed easier in polar coordinates. For detecting the initial contour, a Canny edge detection method ($\alpha = 8$) is used for media-adventitia. After the border detection, the images have to be converted back to the Cartesian coordinates in order to continue edge detection with deformable model and finally to be understood well by the physicians.

Manual processing of IVUS images is a tedious and time consuming procedure. Many efforts have been made in order to develop an accurate automated method for the detection of the regions of interest in IVUS images. Many restrictions in automated segmentation of IVUS images derive from the quality of the image, such as the lack of homogeneity of regions of interest and shadowed regions, which are produced by the presence of calcium [12].

Let us consider a dynamic curve as $X(s,t)=[X(s,t),Y(s,t)]$ where t is the time and s is the curve parameter. Let us also to denote the curve's inward unit normal as N and its curvature as k. The evolution of the curve along its normal direction can be characterized by the following partial differential equation:

$$\frac{\partial X}{\partial t} = V(\kappa)N \tag{1}$$

where *V(k)* is the speed function since it determines the speed of the curve evolution. In the level set method, the curve is represented implicitly as a level set of a 2D scalar function which is usually defined on the same domain as the image itself. The level set is defined as the set of points that have the same function value.

We now derive the level set embedding of the curve evolution. Given a level set function *φ(x,y,t)* with the contour *X(s,t)* as its zero level set we have *φ[X(s,t),t]*. Differentiating this term with respect to *t* and using the chain rule, we obtain:

$$\frac{\partial \phi}{\partial t} + \nabla \phi \frac{\partial X}{\partial t} = 0 \tag{2}$$

where $\nabla \varphi$ denotes the gradient of *φ*, assuming that *φ* is negative inside the zero level set and positive outside it. Equation (2) can be rewritten to (3) according to inward unit normal to the level set curve:

$$\frac{\partial \phi}{\partial t} = V(\kappa)|\nabla \phi| \tag{3}$$

where *k* the curvature at the zero level set is given by:

Since the evolution (3) is derived for the zero level set only, the speed function *V(k)*, in general, is not defined on other level sets. Hence, we need a method to extend the speed function *V(k)* to all of the level sets. A speed function that is used by geometric deformable contours, takes the following form:

$$\frac{\partial \phi}{\partial t} = c(\kappa + V_0)|\nabla \phi| \tag{4}$$

where

$$c = \frac{1}{1 + |\nabla (G_\sigma * I)|} \tag{5}$$

A positive value of V_0 shrinks the curve while a negative V_0 expands it. The curve evolution is coupled with the image data through a multiplicative stopping term. This scheme can work well for objects that have good contrast. However, when the object boundary is indistinct or has gaps like the IVUS image in our case, the geometric deformable contour may leak out because the multiplicative term only slows down the curve near the boundary rather than completely stopping the curve. Once the curve passes the boundary, it will not be pulled back to recover the correct boundary. To overcome this deficiency a new term is added to (5) as shown in (7).

$$\frac{\partial \phi}{\partial t} = c(\kappa + V_0)|\nabla \phi| + \nabla c \nabla \phi \tag{6}$$

The resulting speed function has an extra stopping term $\nabla C \nabla \varphi$ that can pull back the contour if it passes the boundary [11].

2.2 Characterizing Calcified Region

In the previous section, media-adventitia boundary was detected with geometric deformable model method. The region between this border and the catheter is extracted. Vessels' plaques are generally composed of calcium, fibrous and lipid. Calcified regions in IVUS images can be recognized by following characteristics:

- They are usually represented by bright intensity among plaque region.
- As calcium is a hard plaque, the ultrasound beam is not strong enough to penetrate through it. Therefore, the calcified region is usually followed by a dense shadow. Two methods for characterizing calcified regions are examined here. The first one is Bayes classifier and the other is implemented via setting a threshold value on the pixel intensities. For both techniques the results are confirmed by checking the shadow behind the calcified region.

2.2.1 Bayes Classifier

Bayes classifier is based on maximum a posteriori probability, in which the feature vector *X* is assigned to class ω_j if

$$P(\omega_j \mid X) \succ P(\omega_k \mid X) \text{ , } k=1, 2\ldots M, \tag{7}$$

where *M* is the number of classes and *P* is the probability. As the amount of a posteriori probabilities are unknown in most application, its quantity should be calculated through Bayes algorithm where

$$P(\omega \mid X) = \frac{P(X \mid \omega) \times P(\omega)}{P(X)} \tag{8}$$

From the above equation and considering that *P(X)* is equal for all classes, (6) can be written as

$$P(X \mid \omega_j) \times P(\omega_j) \succ P(X \mid \omega_k) \times P(\omega_k) \text{ ,} k=1, 2\ldots M \tag{9}$$

Here we have two classes, one calcified (ω_1) and another non-calcified (ω_2). The Probability Density Function (pdf) of classes is assumed to be Gaussian; therefore, only the values of mean and variance of pdf's should be estimated. These values are attained from 30 images, which were collected by the expert from the dataset, and set as follows: mu_1= 245, mu_2=70, sig_1=20, sig_2=20.

The values of a priori probabilities are achieved experimentally from 30 images and are as follows: P (calcified region)=0.1 , p(Non-calcified region)=0.9.

2.2.2 Thresholding Method

In this method, by studying the pixels' intensities for two classes (calcified and non-calcified) in 30 images, the average value is achieved and is set to 173.50. Therefore, the pixel intensities which are greater than the threshold value are set to calcified class and the others are set to non-calcified class.

2.2.3 Checking the Shadows

In order to increase our classifier reliability, the shadows following the detected calcified regions should get verified and in the case of existing shadow, these identified regions are accepted to be calcified region, otherwise they are set back to non-calcified class. As the IVUS images have circular trait, we first transform the images to polar coordinates where the shadows behind detected calcified regions can easily be checked. For this goal, average of pixels' intensities, which are placed in the same angle as identified regions, is calculated and the one that it's value is below 70, is accepted to remain as calcified class.

3 Result and Discussion

Our study group as sequences of IVUS images includes 7 patients. These images with the digitized matrix size of 500*500 pixels were acquired using a 30-MHz transducer at a 0.5 mm/s pullback speed. 60 frames from each patient have been gathered. The accuracy of the mentioned method is determined by comparing this method resulting borders with borders identified by an expert.

3.1 Border Detection

To put the accuracy of the segmentation into numbers, average distance (AD) and Hausdorff distance, the maximum distance between boundaries, (HD) between automatic and manually traced borders were calculated. These distances directly depict point to point contour variations. The level set based method was applied to each frame to detect intimia and medi-adventitia layers by defining a distance function from the initial border that were detected in preprocessing step. Fig. 1 shows an example outcome of the method.

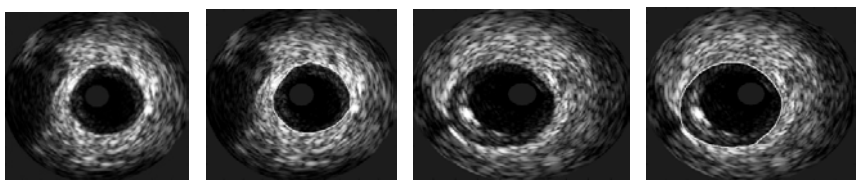


Fig. 1. Normal and calcified original images (number 1 and 3) and the result of method (number 2 and 4)

The cross sectional media_adventitia distances from manually traced and automatically detected boundaries were compared, and the results were expressed as systematic and random differences (Means ± Standard deviation). The automatically determined borders corresponded very well with expert manual measurements. The Average (AD) and Hausdorff distance (HD) values were obtained for the proposed method, demonstrating that this method is powerful for simulated IVUS segmentation (because of the low value for these measurements). These two values for the first expert were 0.344±0.108 and 0.650±0.150 (mm) respectively. For the second expert the values for AD and HD were 0.360± 0.100 and 0.780±0.180 (mm). Area difference value was calculated for the non-overlapping area borders of proposed method and parametric deformable model, the result was 12.3±1.9 and 13.3±2.4 (mm^2) respectively.

Table 1. Performance of Calcified region characterization methods

Classification Method	Sensitivity	Specificity
Bayes Classifier	92.674%	98.5%
Thresholding method	74.14%	83.7%
Adaptive Thresholding [6]	84%	88%
Texturural Features [14]	85.5%	97.2%

3.2 Classification Accuracy

The calcified regions detected by the implemented methods were also compared with the manually detected images. In order to validate the system, the sensitivity and specificity of calcified class were calculated. For this purpose true and false positive (TP and FP), and true and false negative (TN and FN) values were computed and sensitivity and specificities were calculated. The sensitivity for a class is the percentage of members of that class that are correctly classified by the test. As such, it has to be as high as possible. The specificity for a class is the percentage of members of the other classes that are correctly classified by the test. Table 1 shows these values for two classification method. The results of different steps of characterizing calcified region by means of Bayes classifier are demonstrated in Fig. 2.

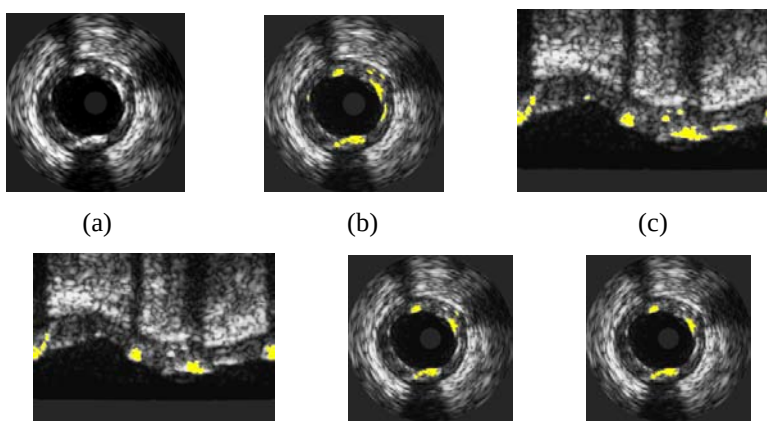


Fig. 2. Characterizing calcified region. (a) IVUS image, (b) High intensity plaque identification by Bayes algorithm (yellow regions indicates calcified area), (c) Transformed image in Polar coordinate, (d) Removing identified plaques which are not followed by shadow , (e) Reconstructed image in Cartesian coordinate (final result), (f) Manually characterized calcified image by expert.

4 Conclusion

Border detection and region identification in IVUS images are a challenging task in medical imaging analysis. Few algorithms have been developed in order to trace media-adventitia border automatically. In this paper the preprocessing includes median filtering that reduces the noise well and preserves the edges of the image and second, detection of the initialize contour with edge detection methods that makes the deformable model method automatic. The initial contour is defined to be a distance function for the evolution equation. After detecting the border, calcified regions are identified with Bayesian classifier and thresholding method. In our validation methodology we compared the results from the implemented methods with the manual estimation of borders by an expert. We observed small variations between manual and automated detection of borders, this denotes that this automatic method was accurate.

Geometric deformable models have some advantages over parametric models. First, they are completely intrinsic and, therefore, are independent of the parameterization of the evolving contour. In fact, the model is generally not parameterized until evolution of the level set function is complete. Second, the intrinsic geometric properties of the contour, such as the unit normal vector and the curvature, can be easily computed from the level set function. We have solved the problem of the places of the nodes in initialization stage for both methods. The proposed method has limitations such that it is not accurate where there is the artifact of existence of other branches or the artifact of curvature of the vessel or catheter and sometimes the frames with calcified shadowing artifacts, for this reason we suggest to use other frames around this frame for decision of the place of the borders.

Also in this paper, the ability of Bayes classifier in characterizing calcified region was investigated. The results of our study show that this method has improved the value of sensitivity and specificity in comparison with other algorithms such as thresholding or using texture based features.

Acknowledgments

The authors would like to thank Mr. Shaki for his suggestions and valuable comments and VOLCANO Company for their help in providing the data set.

References

1. Schoenhagen, P., Stillman, A.E.: Principles, Advances, Clinical Uses of IVUS. Clinic journal of medicine 72, 43–45 (2005)
2. Michailovich, O.V., Tannenbaum, A.: Despeckling of Medical Ultrasound Images. IEEE Trans. on Ultrasonics, ferroelectrics and frequency control 53, 64–78 (2006)
3. Filho, E.D., Yoshizawa, M.: A Study on Intravascular Ultrasound Image Processing. Journal of Mathematical Imaging and Vision 21, 205–223 (2004)
4. Gil, D., Radeva, P., Saludes, J.: Segmentation of Artery Wall in Coronary IVUS Images: A Probabilistic Approach. Computer in Cardiology, 687–690 (2000)

5. Bourantas, V., Plissiti, E., Fotiadis, I., Protopappas, C., Mpozios, V., Katsouras, S., Kourtis, C., Rees, M.R., Mivhalis, K.: In vivo Validation of a Novel Semi-automated Method for Border Detection in Intravascular Ultrasound Images. The British Journal of Radiology 78, 122–129 (2005)
6. Wang, L.Y., Wang, W.: Estimating Coronary Artery Lumen Area with Optimization-based Contour Detection. IEEE Trans. Med. Imag. 22, 48–46 (2003)
7. Papadogiorgaki, M., Mezaris, V., Chatzizisis, Y.S., Kompatsiaris, I., Giannoglou, G.D.: A Fully Automated Texture-based Approach for the Segmentation of Sequential IVUS Images. In: International Conference on Systems, Signals & Image Processing, vol. 8, pp. 461–464 (2006)
8. Cardinal, M., Meunier, J., Soulez, G., Maurice, R.L.: Intravascular Ultrasound Image Segmentation: A Three-dimensional Fast-marching Method Based on Gray Level Distributions. IEEE Trans. Med. Imag. 25, 590–601 (2006)
9. Brathwaite, P.A., Chandran, K.B., McPherson, D.D., Dove, E.L.: Lumen Detection in Human IVUS Images Using Region-growing. IEEE Conference in Cardiology 8, 37–40 (1996)
10. McInerney, T., Terzopoulos, D.: Deformable Models in Medical Image Analysis: A Survey. Med. Imag. Anal. 1, 91–108 (1996)
11. Han, X., Xu, C., Joil, P.: A Topology Preserving Level Set Method for Geometric Deformable Models, vol. 25, pp. 755–768. IEEE Computer Society, Los Alamitos (2003)
12. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active Contour Models. Int. J. Comp. Vision 1, 321–331 (1987)
13. Vince, D.G., Dixon, K.J., Cothren, R.M., Cornhill, J.F.: Comparison of Texture Analysis Methods for the Characterization of Coronary Plaques in Intravascular Ultrasound Images. Computerized Medical Imaging and Graphics 24, 221–229 (2000)
14. Brunenberg, E.J.L.: Automatic IVUS Segmentation Using Feature Extraction and Snakes. Internship report, Dept. of Biomedical Engineering, Eindhoven University of Technology, The Netherlands (2005)

Object Modeling for Multicamera Correspondence Using Fuzzy Region Color Adjacency Graphs

Amir Hossein Khalili and Shohreh Kasaei

Sharif University of Technology, Azadi Ave,Tehran, Iran
a_khalili@ce.sharif.edu, skasaei@sharif.edu

Abstract. In this paper, a novel moving object modeling suitable for multicamera correspondence is introduced. Taking into consideration the color and motion features of foreground objects in each independent video stream, our method segments the existing moving objects and constructs a graph-based structure to maintain the relational information of each segment. Using such graph structures reduces our correspondence problem to a subgraph optimal isomorphism problem. The proposed method is robust against various resolutions and orientations of objects at each view. Our system uses the fuzzy logic to employ a human-like color perception in its decision making stage in order to handle color inconstancy which is a common problem in multiview systems. The computational cost of the proposed method is made low to be applied in real-time applications. Also, it can solve the partial occlusion problem more precisely than the Meanshif occlusion solver by 15.7%.

Keywords: Object Correspondence, Multicamera Tracking, Region Adjacency Graph, Fuzzy Color Modeling.

1 Introduction

The increasing demand for analyzing moving objects behavior in wide area has heightened the need for robust modeling and correspondence of the moving objects in scene. The purpose of object modeling is taking out some features from images of an object so that the selected features are stable and reliable and could exactly discriminate the target in consequence projections.

For applications such as outdoor video surveillance, where the entire scene is not coverable with a single camera, a distributed camera network should be implemented. While tracking in multi cameras network, selected features should have little changes from different views, so that specific object could be discriminate easily from other cameras to track. Variation in cameras used in a network and their different positions, directions and distances to target, in addition to different illumination conditions cause selection of stable features in different views to be a challenging problem. Changes in size, view direction, luminance and color values of objects are some artifacts. For that reason, special object model is needed to maintain correspondence of two-dimensional projections of an object seen in different views.

Wide variety of methods has been reported to model moving objects in multi-view applications, but a few of them handle color variation and occlusions effectively

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 637–644, 2008.

without employing high-level reasoning procedure and predefined model of target objects. In our experience the low level image features play a crucial role.

The rest of paper is organized as follows. In section 2 an overview on related previous works is given. In section 3 different steps of the proposed algorithm are introduced. The experimental results are shown in section 4. Finally, section 5 concludes the paper.

2 Previous Work

During tracking with single camera, objects could be modeled according to their temporal status information including position, temporal velocity, appearance features and color properties[1]. Most multicamera applications use color information of target objects as modeling parameters. [2] used average color of moving objects in HSV space. [3] use only one color histogram model per object. Although color histograms are rotation/scale invariant, computationally efficient and robust to partial occlusion, but do not consider distribution of colors. Reference [4] tried to find relation of sensed color through a training step. [5] used Munsell color space [6] for constructing histograms. In this approach colors are coarsely quantized into 11 predefined bins. [7] used color-spatial distribution of moving object by partitioning moving blob in its polar representation. Reference [8] claims that color information of body, pelvis, and feet are the most stable information. In [9] authors cluster a person according to its color and motion features using watershed and K-means algorithm to recognize gestures and track each.

One approach to the detection and tracking problem is to fit explicit object models of shape [10]. Some model fitting approaches focused on high-level reasoning [11]. The robustness of such approaches is highly depends on the defined model. Models cover limited range of objects and behaviors and fails if the moving objects move fast or don't obey the definite behaviors. For such methods calibration data of cameras should be known. Occlusions, low resolution images and variety of poses and colors which are common in most applications are some crucial peril for robustness of these methods.

3 Proposed Method

3.1 Moving Object Segmentation

Background subtraction method [12] used for extracting foreground regions in each camera. Color information and direction of motion vectors are two features are used for object segmentation. In this regard we used Horn & Schunck algorithm and Cr and Cb channels of YCrCb color space. The advantage of using motion vectors in segmentation processes is that the direction and distribution of the optical flow can be used to distinguish the different moving parts of objects. Even though these parts may have similar color in one view, they usually differ in others. Therefore, no extra process is needed to split and merge different part of object model to match them in different views. In addition, motion is usually a reliable feature to detect human activity in indoor environment and from different outlooks [9].

Before segmentation a preprocessing is needed to smooth isolated pixels which their color or optical flow don't confirm with their neighbors. We suggest using bilateral smoothing filter [13] which is a simple, non-iterative scheme for edge-preserving smoothing. It replaces the pixel value at x with an average of similar and nearby pixel values and doesn't introduce false colors around the boundaries. By implementing such filter 3-5 times, we obtain graphic like appearance in which fine texture has gone.

We adopt the approach introduced in [14] to segment foreground regions because of three major reasons: 1) considering global characteristics of pixels. 2) Low computational complexity. 3) The graph-based structure of this algorithm highly conforms to our region adjacency graph (RAG) model. In this sense root nodes of segmentation algorithm which represent different regions correspond to nodes of RAGs. Two nodes in RAG are adjacent if their corresponding nodes in segmentation algorithm are adjacent, Fig. 1.

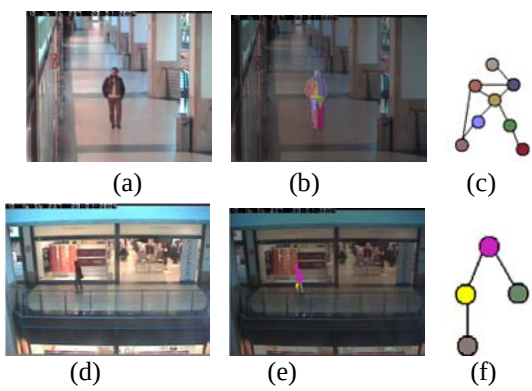


Fig. 1. Result of proposed segmentation in a natural scene. a) Original corridor frame. b) Foreground segmentation of a. c) Region adjacency graph of b. d) Original frame of front view. e) Foreground segmentation of d. f) Region adjacency graph of e.

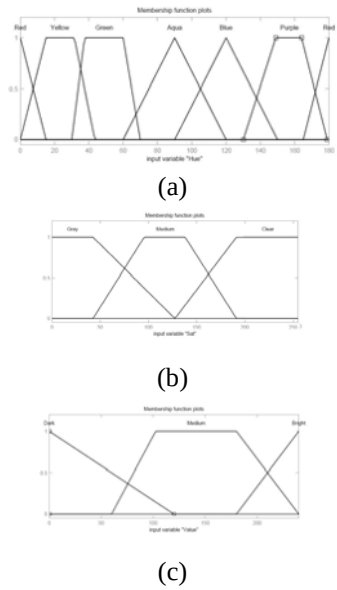


Fig. 2. a.,b,c: Fuzzy sets of *Hue*s, *Saturation* and*luminance*

3.2 Human Perception-Based Color Modeling

In multiview applications human classifies colors to a limited set which is more stable. Mapping colors of object projection in a view to members of this set, human looks for anyarrangement of colors in other views that confirms with his target model. Human color classification is not course. He some times faces a dilemma about a color. In such a situation human assigns a membership value to each perceptual color.

Course partitioning the color space into a relatively small number of course segments such as [6] does not provide an optimal solution for correspondence problem. To help our system to have humanoid color classification, we used HSL color space and fuzzy logic systems. The antecedent variables of rules are *Hue*, *Saturation* and *luminance*. The membership function of them are shown in Fig. 2. The consequent part of each fuzzy rule is a crisp discrete value of the set *Black*, *White*, *Red*, *Orange*, *Yellow*, *Gray*, *Brown*, *Aqua*, *Blue*, *Green*, *and Purple*. The members of this set are called *Linguistic* colors.

By defining set of fuzzy rules according to human belief and observation, perception of linguistics colors based on antecedent variables are translated to system, so that system could inference the linguistic colors and their membership believes similar to the human. For example, the rule "*Yellow* $\wedge$ *Medium* $\wedge$ *Medium* $\rightarrow$ *Brown*" is defined by setting H = 25, S = 128, L =150 (points of maximum of the membership functions) and observing human belief about generated color. There are totally 64 rules. The reasoning procedure is based on a zero-order Takagi-Sugeno model. In proposed method, base on H, S, L values of a pixel membership beliefs to each linguistic color is computed. Accumulative beliefs of region's pixels to each 11 linguistic colors, construct a histogram. After dividing each cell of the histogram to area of the region, the it is called *linguistic color histogram.* For each node of RAG a linguistic color histogram is associated.

3.3 Matching Two Fuzzy Region Adjacency Graphs

Similarity between two objects modeled in term of RAGS can be approximated by measuring subgraph optimal isomorphism of their graphs. Also, thanks to subgraph optimal isomorphism we can discriminate RAGs of different objects in case of partial occlusion. We approximate this similarity measurement by adapting method introduced in [15]. According to [15] it is better to decompose the two candidate RAGs to smaller subgraphs called Basic Attributed Relational Graphs (BARG). A BARG is one level tree, including a node (denoted by *n*) and all its neighbors. Fig. 3 shows different BARGs of presented RAGs in Fig 1.d.

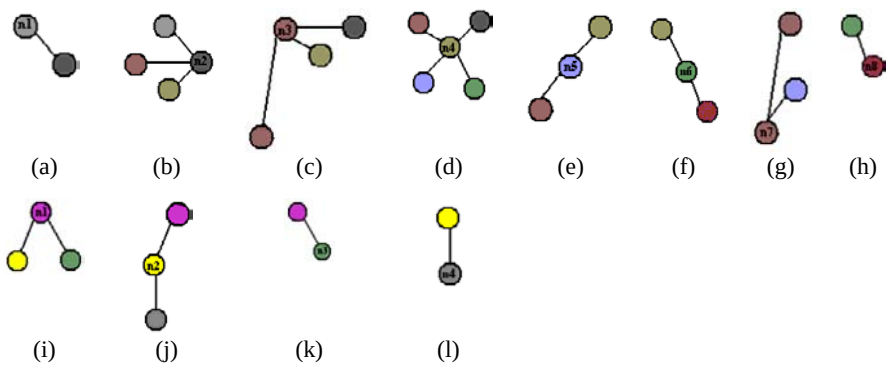


Fig. 3. (a-h) Basic attributed graphs of region adjacency graph of Fig. 1.c. (i-l) Basic attributed graphs of region adjacency graph of Fig. 1.e.

Subgraph optimal isomorphism measurement can be calculated with optimal matching of a complete weighted bipartite graph whose nodes are BARGs of each RAG partitioned in two groups. Weight of connecting edge between i'th BARG of RAG v in camera C1 and j'th BARG of RAG u in camera C2, is computed using (1).

$$\underset{c1,c2}{Dist}(v_i,u_j)=\underset{c1,c2}{Dist}(n_{v_i},n_{u_j})+w_n\times\left|\deg(n_{v_i})-\deg(n_{u_j})\right|+w_e\times\left|\deg(n_{v_i})-\deg(n_{u_j})\right| + \underset{c1,c2}{Dist}_{bipartite}(N(n_{v_i}),N(n_{u_j}))\,. \tag{1}$$

$$\underset{c1,c2}{Dist}(n_{v_i},n_{u_j})=\sum_{y,x\in\ linguistic colors}(w_{x,y}^{c1,c2}\times\min(n_{v_i}.hist(x),n_{u_j}.hist(x)))^2$$

$w_{x,y}^{c1,c2}$ is a coefficient which introduces similarity between two linguistic colors in two cameras. For example It is likely to visit a gray color in one view as black one in another view. So we define $w_{gray,black}=0.5$. It is possible to obtain the coefficients according to seen colors of definite moving object in a training step.

When two regions have great number of pixels in common at specific linguistic color, formula (1) increases their similarity measurement more significant than the status in which they have same amount of intersection but distributed in more linguistic colors.

$\deg(n_{v_i})$ in (1) denotes degree of root of *i*th BARG of RAG *v*. w_n and w_e denote node and edge insertion costs. It is better to define w_n and w_e adaptive and as inverse function of difference in size of foreground objects. In this manner we can efficiently handle different resolutions of object in different cameras. In case of occlusion, w_n and w_e don't change. $Dist_{bipartite}(N(n_{v_i}),N(n_{u_j}))$ is calculated as the maximum matching of a weighted bipartite graph constructed from neighbors of n_{v_i} and n_{u_j} in each its part.

For tracking objects in each camera we use combination of sequential Kalman filter and joint probabilistic data association (JPDA) [16]. Similarity measurement introduced in section 3.3 is used to find most feasible hypothesis in validation gate of JPDA. For tracking in single camera we set $w_{x,y}=1$ iff $x=y$ and $w_{x,y}=0$ if $x\neq y$. It is due to the fact that we have not color distortion in a single view. In case of occlusion, RAG of foreground in place of occlusion is called Super Region Adjacency Graph (SRAG). In case of occlusion, the RAG which can be segmented completely from SRAG is RAG of most closed objects. By removing founded RAGs from SARG we can do subgraph isomorphism process again to locate other objects.

4 Experimental Results

To evaluate usefulness of our proposed method, we implemented the multi object tracking system using OpenCV infrastructure. We run our code on 3.00 GHZ system with 2.00 GB RAM. we used CAVIAR [17] and PETS2001[18] (resized to 384×260)

as our datasets which are outdoor streams. Fig. 4 and Fig. 1 (section 3.1) show our results obtained from 2 pair frames of "OneStopNoEnter2" sequences in CAVIAR. Fig. 5 shows three consequent frames of "TwoEnterShop1" sequence of CAVIAR. It is noticeable how well different parts of the frontier person are segmented. However, by going far and thus decreasing in resolution of moving objects the accuracy is reduced, but we still obtain meaningful segments.

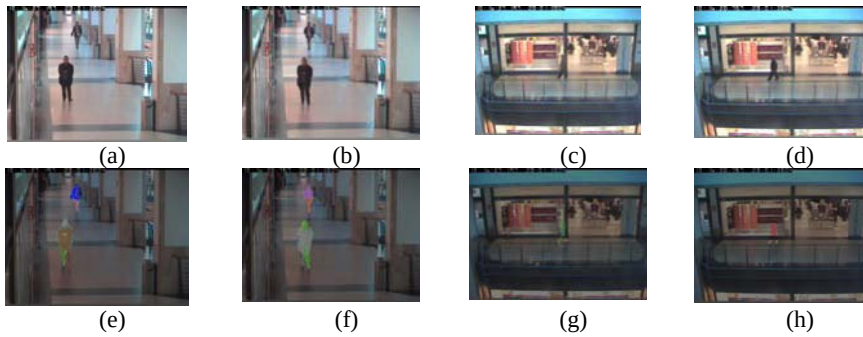


Fig. 4. (a-b) Original frames of corridor view. (c-d) Original frame of front view. (e-f) Our segmentation results for (a-b). (g-h) Segmentation results of (c-d).

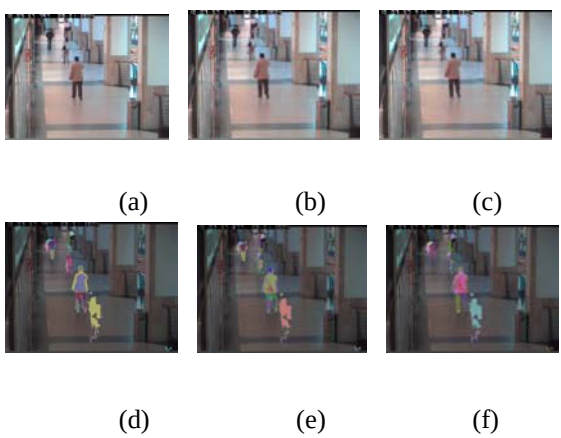


Fig. 5. Three consequent frames of "*TwoEnterShop1*" sequence of CAVIAR along the corridor view

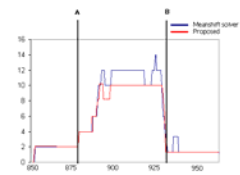


Fig. 6. Error of occlusion solvers between frames 888 and 937: *blue*: Meanshift method, *red*: proposed method

The abnormally segments in the middle of 5.d and 5.e are due to false background segmentation which appears because the initially existence of the human in streams. The proposed method reduces ratio of false and miss matches to total matches by 65%.

Fig. 6 shows the tracking error of selected tracks from the first PETS2001 video sequence. The blue and red colors show the error of occlusion solver using the

Meanshift and proposed method, respectively. The error is calculated using Euclidean distance between center of blob being tracked and center of object yield manually. It can be observed that the largest errors occur when the object splits into several segments due to occlusion. Our proposed method has reduced the error by about 15.7%. Table 1 compares complexity cost of our proposed method with [9].

Table 1. Performance comparison of different moving object segmentation methods

Video Type		Frame Size	Average elapsed time for each frame in seconds.	
			Method Proposed in [9]	Proposed Method
CAVIAR	TwoEnterShop1 corridor view	384×288	1.5167	0.0613
	TwoEnterShop1 front view	384×288	0.0634	0.0500
	OneStopNoEnter2 corridor view	384×288	1.3231	0.0827
	OneStopNoEnter2 front view	384×288	0.0534	0.0500
PETS 2001	TestSet1 Camera1	384×260	0.3458	0.0603
	TestSet1 Camera2	384×260	0.5765	0.0701

5 Conclusion

In this paper, we proposed a novel method to model moving objects in multicamera systems. Our proposed method constructs a region adjacency graph for each foreground object and solves the correspondence problem through a subgraph optimal isomorphism approach. A Fuzzified color histogram is then associated to each graph node as its attribute. Our proposed method is fast enough to be employed in real-time applications. Also, our proposed method is robust to face partial occlusion, color changes, and different object resolutions and orientations.

Acknowledgment. This work was in part supported by a grant from ITRC.

References

1. Velastin, S.A., Remagnino, P.: Intelligent distributed video surveillance system. IEE press, London (2005)
2. Kettnaker, V., Zabih, R.: Bayesian Multi-Camera Surveillance. In: Conf. on computer vision and pattern recognition, pp. 2253–2261 (1999)
3. Orwell, J., Remagnino, P., Jones, G.A.: Multiple Camera Color Tracking. In: IEEE Intl. Workshop on visual surveillance, pp. 1355–1360 (2003)
4. Javed, O., Rasheed, Z., Shafique, K., Shah, M.: Tracking Across Multiple Cameras with Disjoint Views. In: 9th IEEE Int. Conf. on computer vision, Nice, France, pp. 952–957 (2003)

5. Gilbert, A., Bowden, R.: Incremental Modeling of the Posterior Distribution of Objects for Inter and Intra Camera Tracking. In: British machine vision conference, Oxford, UK (2005)
6. Blackman, S., Popoli, R.: Design and analysis of modern tracking systems. ArtechHouse (1999)
7. Kang, J., Cohen, I., Mediono, G.: Continuous tracking within and across camera streams. In: IEEE Conf. on computer vision and pattern recognition, vol. 1, pp. 267–272 (2003)
8. Zajdel, W., Krose, B.J.: A sequential Bayesian algorithm for surveillance with non-overlapping cameras. J. pattern recognition and AI 9, 977–996 (2005)
9. Aghajan, H., Wu, C.: Layered and collaborative gesture analysis in multi camera networks. In: Int. Conf. on acoustics, speech and signal processing, USA, vol. 4, pp. 1377–1380 (2007)
10. Frank, M., Haag, H., Kollnig, H., Nagel, H.H.: Tracking of occluded vehicles in traffic scenes. In: Buxton, B.F., Cipolla, R. (eds.) ECCV 1996. LNCS, vol. 1065, pp. 485–494. Springer, Heidelberg (1996)
11. Ramanan, D., Forsyth, D.A., Zisserman, A.: Tracking people by learning their appearance. Trans. On pattern analysis and machine intelligence 29(1) (2007)
12. Darabi, A., Khalili, A.H., Kasaei, S.: Graph-based segmentation of moving object. In: Int. GCC IEEE. Manama. Kingdom of Bahrain (2007)
13. http://www.dai.ed.ac.uk/CVonline/LOCAL_COPIES/MANDUCHI1/Bilateral_Filtering.html
14. Felzenswal, P.F., Hutencheler, D.T.: Efficient GraphBased Image Segmentation. J. CV 32 (2004)
15. El-Sonbaty, Y., Ismail, M.A.: A new algorithm for subgraph optimal isomorphism. Trans. On pattern recognition 31, 205–218 (1998)
16. Bar-Shalom, Y., Li, X.: Multitarget-multisensor tracking: principles and techniques. YBS publishing (1995)
17. http://homepages.inf.ed.ac.uk/rbf/CAVIAR/
18. http://peipa.essex.ac.uk/ipa/pix/pets/PETS2001/DATASET1/

Secure Digital Image Watermarking Based on SVD-DCT

Azadeh Mansouri, Ahmad Mahmoudi Aznaveh, and Farah Torkamani Azar

Electrical and Computer faculty, Shahid Beheshti University, Tehran, Iran
{a_mansouri,a_mahmoudi,f-torkamni}@sbu.ac.ir

Abstract. Singular Value Decomposition (SVD) has been used as one of the best transform for implementing robust digital watermarking. In some SVD based watermarking techniques, eigenvalues of the watermark are embedded into eigenvalues of the cover image and the eigenvectors are used as key parameter. This category of methods is not capable to provide the secure watermarking technique as the original eigenvectors have been used in extraction process. Apart from that, the other SVD based watermarking which are considered as secure techniques are not enough imperceptible. In this paper, two new secure methods of non-blind image watermarking is performed. These methods are implemented using modification on singular values of the host data by embedding DCT coefficients of the watermark image. Due to the nice properties of DCT as a reversible transform and taking the advantages of same variation as the Singular values change, these coefficients of watermark has been used as the information which is embedded to provide the secure schema. Beside enough robustness against large amount of attacks, the best quality for watermarked image can be achieved by altering just brightness instead of changing the local variation of the signal.

Keywords: Watermarking, Singular Value Decomposition, Data Hiding.

1 Introduction

Digital data utilization along with the increased popularity of the internet has facilitated information sharing and distribution. However, such applications have also raised concerns about copyright issues, unauthorized modification and distribution of digital data such as digital images, audios and videos. Digital watermarking is proposed to solve these problems. Every watermarking technique suggested for copyright protection should satisfy the robustness and imperceptibility requirements and the capacity should be high unless it reduces the ability of technique to satisfy the other two requirements.

The watermark can be applied to spatial domain [1], [2] and [3] or transform domain [4-6]. Modifying the singular value decomposition of the image is one of the most important techniques in transform domain watermarking [7-9]. Considering the previous works, some authors combined SVs of the original image with SVs of the watermark in different ways [6], [7], [8] and [9].

A digital image A of size $M \times N$, when $M \geq N$ can be represented by its SVD as:

$$A = USV^T \tag{1}$$

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 645–652, 2008.

Where $U \in F^{M\times M}$ and $V \in F^{N\times N}$ are unitary matrices and $S \in F^{M\times N}$ is a diagonal matrix, with singular values which are ordered in the original diagonal matrix. Later, this matrix is called SV matrix.

Most of the previous SVD-based watermarking techniques are tried to take the advantages of little degradation of luminance component when it is altered. This arises from the fact that slight changes on singular values of an image do not affect much on the image quality. In this case, in traditional SVD based method, the SVs of the watermark are embedded into SVs of the cover image using variant or non variant scaling factors. Although the result of these combination lead to have the little alternation on watermarked image quality, this category of methods are not capable of providing reliable and secure techniques for watermarking algorithms specially those which are claim of proposing the ownership protection [8].

Regarding these methods, the extracted watermark is constructed using original singular vector. These U and V matrices show the local variances, selecting this information as a secret key provides the situation for the hackers who claim about their watermark image while another image has been embedded in the host as the right watermark. So the SVD based watermarking techniques which use the eigenvalues as embedded information cannot be considered as a method with favorable security. To solve this method's security weaknesses, some modifications on this algorithm have been proposed recently. The distinctive point between these modified methods as described in [12], [13] and the inefficient traditional SVD methods is idea of embedding U or V matrices of the watermark also as a control parameter. At the extraction step, these algorithms also use original U and V to reconstruct watermark image, and just the similarity between the extracted U or V and original ones are checked. U and V preserve the major information about an image but these matrices are noise like signals and very sensible. So it is clear that little altering in watermarked image may lead to remarkable changes in U and V values. Although the experimental results in [12] and [13] show favorable outcomes about the respective U and V similarity check, it should be noticed that the experiments have been conducted just on completely different images and the approved results may not be achieved in case of similar images. Moreover, in all acceptable secure watermarking algorithms, the embedded information has been chosen among the features which contain the major information of the watermark and also in extraction phase these information must be used in reconstructing process. So, the visual extracted watermark which is achieved by using the improved method given in [12] and [13] is not acceptable. In this case the large amount of the extracted watermark's quality improvement is related to the original U and V containing the great amount of information which has been used in watermark reconstruction process improperly. Therefore, these groups of SVD based algorithm can not provide secure watermarking techniques; especially in the field of ownership protection [14], [15] and [16]. We introduce secure SVD based methods which are used the DCT transform of the watermark image to embed them in the SVs of the host. This paper is organized as the following manner. In section II, generally the SVD watermarking technique has been described. Then the details of the new proposed method of image watermarking have been presented. Section III is dedicated to analyze the results of the performed implementation. Finally we conclude this paper in section IV.

2 The Proposed Methods

Selecting the SVs of the watermark as a feature which preserves the major information is not a suitable choice and it may ends in getting false positive results. So in our proposed algorithm DCT coefficients of the watermark image has been selected as the embedding information. These coefficients have the same fluctuations as the SVs value change (Fig.)

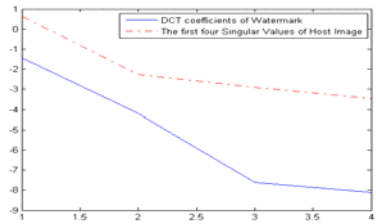


Fig. 1. Variation of the DCT coefficients and four important SV Values(dash line)

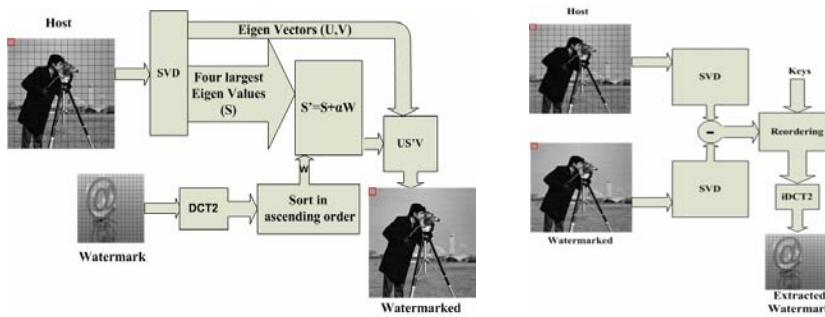


Fig. 2. a) Block diagram of embedding (Proposed Algorithm I) b) Extraction of watermark

Noticing this point, high quality of watermarked image is expected to be realized by our new algorithms. In addition, by using this reversible transform the security problem of traditional SVD based watermarking has been solved. In this case, we propose two different DCT_SVD watermarking algorithms. Each of these methods provides a secure procedure in case of non blind schema. The embedding diagram which is related to proposed algorithm I is shown in Fig.-a. At the first step, blocking process of size m×m is performed on the host image and then SVD is applied in each block. Also, the watermark image is decomposed into h×h block size. Then their DCT transform of blocks were computed and combined by the SVs of each host block separately. In this case, $h \leq \sqrt{m}$ shows the relationship between block sizes and also illustrates the capacity of the data which can be embedded using this algorithm.

The extraction phase would be in inverse order as shown in Fig.-b. Based on the size of the host and watermark block, different quality of extracted watermark and watermarked image can be achieved. To compare the proposed method with the secure SVD based watermarking algorithm, the block based method which is given in

[9] has been selected. In This method the watermark image has been considered in spatial domain. Each pixel of the watermark has been combined with the largest singular value of each 8×8 host block. To show the beneficial point of using DCT coefficients, we implement another version of DCT-SVD method which is named as proposed method II by combining DCT coefficient of watermark with the largest singular value of each 8×8 host block. In this method at first we set some priority on each host block based on its largest singular value. The embedding diagram of the proposed algorithm II is shown in Fig..

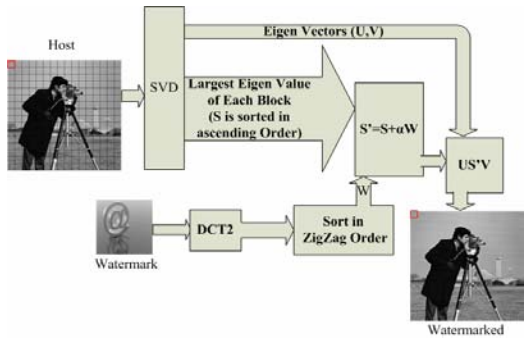


Fig. 3. Block diagram of embedding (Proposed Algorithm II)

The extraction steps have been implemented in reverse order but it should be noticed that in this method two secret key must be defined. One of these keys is related to the priority which is set to each host block and the other one should be generated as a sign of the DCT coefficients of the watermark image.

3 Experimental Results

At the first step, a number of experiments have been conducted using grayscale watermark. To substantiate our methods we have used cameraman of size 512×512 and "@" logo of size 64×64 as the host image and the watermark respectively. We perform the simulations for two DCT methods which have been described in previous section in comparison with the existing secure block based SVD method [9]. To achieve better performance four largest values among 16 SVs of each 16×16 host image block has been selected and four DCT values of each 2×2 watermark block have been chosen.

In the proposed algorithm II, the DCT coefficients of watermark of size 64×64 has been scanned in zigzag order and then these values have been used as embedding data. The performance of the watermarking process is highly depending on choosing the proper scaling factor. In this case, the greater scaling factor leads to gain more degraded watermarked image. It should be considered that the correlation between the extracted watermark and the original one varies based on the selected scaling factor. There is always a trade off between the correlation coefficient of the extracted watermark image and the PSNR value of watermarked data. The same value as scaling factor in both the proposed method I and the block based SVD method [9] have been

selected. But in the proposed method II the variant scaling factors have been chosen based on the block priority. These values will be increased in linear manner. The scaling factor parameter will be grown up as $\alpha_k = \alpha \times B_k / N$, which α_k indicates the appropriate scaling factor corresponding to the k^{th} block and N shows the total number of blocks. To compare the watermarked quality, the PSNR value of the watermarked image is illustrated in Table.

Table 1. Comparison between the PSNR Value of Watermarked Image in three methods

Method	$PSNR = 10\log\frac{255}{MSE}$
Proposed Method I	36.98dB
Proposed Method II	44.85dB
SVD based Method[9]	36.91dB

3.1 Applying Attack

In the following, the proposed methods will be tested against wide range of common attacks. In Fig. all the watermarked images before and after some non-malicious attacks have been shown. Fig. , Fig. and Fig. 7. show the extracted watermark related to different kinds of image manipulation of the three methods.

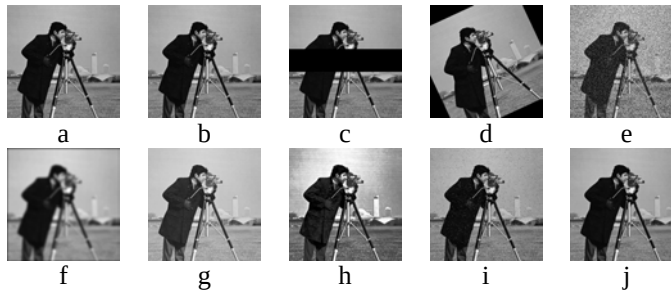


Fig. 4. Different attack on the watermarked image, a) Without attack, b) Compression attack (JPEG 75%), c) Cropping, d) Rotation, e) Gaussian Noise, f) Gaussian Blur, g) Gamma Correction, h) Histogram equalization, i) Salt & Pepper Noise, j) Median filter

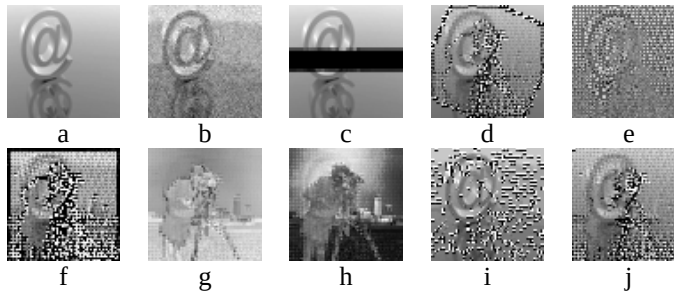


Fig. 5. Extracted watermarked with different attack in the proposed method I, corresponded to each part of Fig

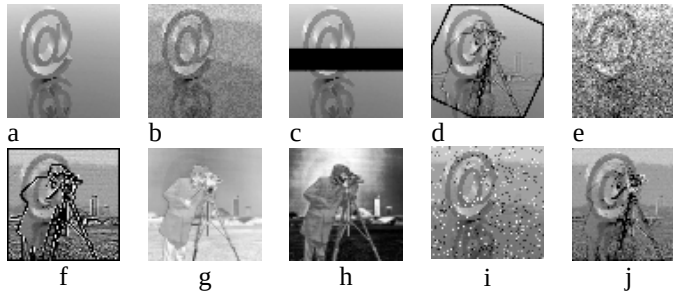


Fig. 6. Extracted watermarked with different attack in the Block SVD based method, corresponded to each part of Fig.

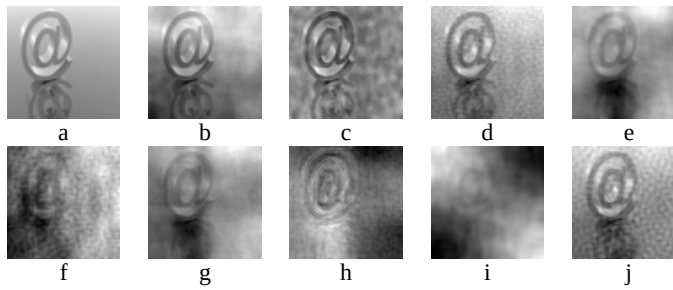


Fig. 7. Extracted watermarked with different attack in the proposed method II, corresponded to each part of Fig.

3.2 Quality Measurement

Previous figures depict the result of the extracted watermark visually which all comparable subjectively. But in all watermarking systems, it is recommended to compare the result objectively using the acceptable visual quality metric.

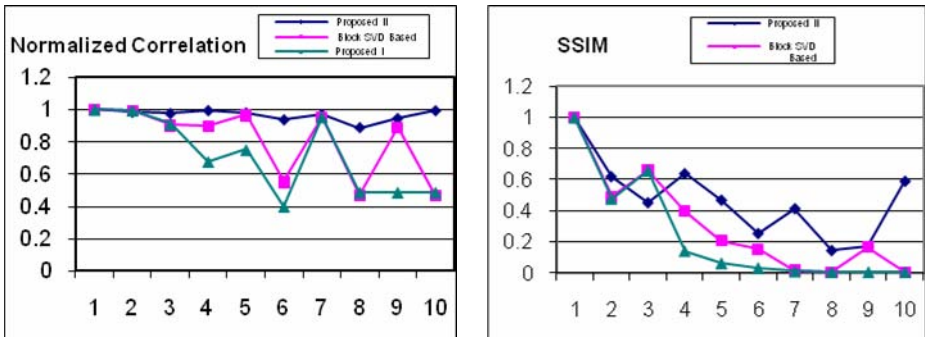
To achieve fair comparison and also coherent performance evaluation, selecting the appropriate quality assessment method is very important [10]. Since in many cases the poor result of the numerical metric may be qualified by subjective test (which is evaluated by humans) [11], in addition of the common metric like Normalized correlation, which is shown in equation (2), we have tested the results with SSIM method as one of the state of the art quality metric [10].

$$\rho(\mathbf{W},\hat{\mathbf{W}}) = \frac{\sum_m \sum_n w_{mn}\hat{w}_{mn}}{\sqrt{\sum_m \sum_n w^2_{mn}}\sqrt{\sum_m \sum_n \hat{w}^2_{mn}}} \tag{2}$$

The extracted results of the two proposed methods using the two types of quality metrics are shown in Fig.-a and b. In the mentioned figures each number is corresponding to the attack which is depicted in Fig. .

3.3 Binary Watermarks

In addition to the extracted watermark quality and also the PSNR of the watermarked image which is described in the previous section, another privilege of our method can be depicted in visual watermarked quality specially in case of embedding binary watermark. The results of these implementations are shown in Fig..



a b

Fig. 8. Comparison between the extracted watermarks of the three methods, a) using normalized correlation b) using SSIM

It is clear that the effect of the watermark can be seen easily in the block SVD based methods. For fair comparison, the scaling factors have been set in such a way that it provided the same PSNR values for the watermarked image using three methods.

Fig. 9. a) Original watermark, Watermarked image: b) Block SVD Based methods c) Proposed method I d) Proposed method II, difference between watermarked and host image: e) Block SVD Based methods f) Proposed method I g) Proposed method II

4 Conclusion

Most of the traditional SVD based watermarking technique take the advantages of using the eigenvalues as the embedding information. These kinds of watermarking algorithms can not be considered as class of methods with favorable security. To achieve a set of secure methods using the beneficial points of SVD based watermarking, modification on eigenvalues have been suggested to be applied just in the host image. Altering the SVs value of the host image cause little degradation in watermarked data.

In order to embed watermark in a secure manner, a reversible transform has been selected. Due to the remarkable properties of DCT transform and taking the advantages of having the same fluctuations like the SVs values, at least the same watermarked quality as the traditional SVD watermarking algorithm would be achieved. Furthermore, the extracted watermarks are robust against common attacks. Although the DCT coefficients of watermark signal provide secure schema, the additional privilege of our algorithm is its imperceptibility, which has been achieved by little alternation in brightness component.

References

1. Darmstaedter, V., Delaigle, J.F., Quisquater, J.J., Macq, B.: Low Cost Spatial Watermarking. Computers & Graphics 22(4), 417–424 (1998)
2. Kimpan, S., Lasakul, A., Kimpan, C.: Adaptive Watermarking in Spatial Domain for Still Images. In: Proc. of the Int. Conf. on Information and Knowledge Engineering, pp. 32–137 (2004)
3. Nikolaidis, N., Pitas, I.: Robust Image Watermarking in the Spatial Domain. Signal Processing (EURASIP) 66(3), 385–403 (1998)
4. Cox, I.J., Kilian, J., Leighton, T., Shamoon, T.: Secure Spread Spectrum for Multimedia. IEEE Trans. on Image Processing 6(12)(1997)
5. Guo, H., Georganas, N.D.: Digital Image Watermarking for Joint Ownership Verification without a Trusted Dealer. In: Proc. IEEE ICME 2003, Baltimore, USA (2003)
6. Hu, A.T.S., Chow, A.K.K., Woon, J.: Robust Digital Image-in-Image Watermarking Algorithm Using the Fast Hadamard Transform. In: IEEE Int. Symposium on Circuits and Systems, Thailand, vol. 3 (2003)
7. Ganic, E., Zubair, N., Eskicioglu, A.M.: An Optimal Watermarking Scheme Based on Singular Value Decomposition. In: Proc. of the IASTED Int. Conf. on Communication, Network, and Information Security (CNIS 2003), Uniondale, NY (2003)
8. Liu, R.Z., Tan, T.N.: SVD-Based Watermarking Scheme for Protecting Rightful Ownership. IEEE Trans. on Multimedia 4(1) (2002)
9. Chandra, D.V.S.: Digital image watermarking using singular value decomposition. In: Proc. of 45th IEEE Midwest Symposium on Circuits and Systems, Tulsa Oklahoma (2002)
10. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image Quality Assessment: From Error Measurement To Structural Similarity. IEEE Trans. on Image Processing 13(4) (April 2004)
11. Wu, H.R., Rao, K.R.: Digital Video Image Quality and Perceptual Coding. CRC Press, Taylor & Francis group (2006)
12. Yavuz, E., Telatar, Z.: Improved SVD-DWT based digital image watermarking against watermark ambiguity. In: ACM symposium on Applied computing, Seoul, Korea, pp. 1051–1055 (2007)
13. Yavuz, E., Telatar, Z.: SVD Adapted DCT Domain DC Subband Image Watermarking Against Watermark Ambiguity. In: Gunsel, B., Jain, A.K., Tekalp, A.M., Sankur, B. (eds.) MRCS 2006. LNCS, vol. 4105, pp. 66–73. Springer, Heidelberg (2006)
14. Wu, Y.: On the security of an SVD-Based Ownership Watermarking. IEEE Transaction of Multimedia 7(4), 624–627 (2005)
15. Zhang, X.P., Li, K.: An SVD-Based Watermarking Scheme for Protecting Rightful Ownership. IEEE Transaction of Multimedia 7(2), 593–594 (2005)
16. Ting, G.C.: Ambiguity Attacks on the Ganic-Eskicioglu Robust DWT-SVD Image Watermarking Scheme. In: Won, D.H., Kim, S. (eds.) ICISC 2005. LNCS, vol. 3935, pp. 378–388. Springer, Heidelberg (2006)

A Novel Delay Fault Testing Methodology for Resistive Faults in Deep Sub-micron Technologies

Reza Javaheri and Reza Sedaghat

Department of Electrical and Computer Engineering,
Ryerson University, Toronto, ON, Canada
{rjavaher,rsedagha}@ee.ryerson.ca

Abstract. Delay failures are becoming a dominant failure mechanism in nanometer technologies. We propose an algorithm for gate-delay fault diagnosis in deep sub-micron by a series of injections and evaluations to propagate the actual timing faults as well as those delay faults that eventually create logic faults to the primary outputs. Unlike the backtrack algorithm that predicts the fault site by tracing the syndrome at a faulty output back into the circuit, this approach propagates the fault from fault site by mapping a nine-valued voltage model on top of a five-valued voltage model. In such a forward approach, accuracy is much higher as all the composite syndromes at all faulty outputs are considered simultaneously. As a result, the proposed approach is applicable even when the delay size is relatively small. Experimental results show that the number of fault candidates produced is considerable.

Keywords: Resistive Short and Open Faults, MVL5 and MVL9, Delay.

1 Introduction

Timing failure analysis is a procedure used to locate the source of timing failures. Unfortunately, existing delay-fault diagnosis methodologies suffer from poor resolution or low scalability. When a physical defect leads to excessive delays on signals instead of altering the logic function of the circuit it is no longer a static defect. Such unknown and unpredictable defect behaviors make it very difficult to analyze a fault. The dynamic nature of such faults disturbs the timing of the logic propagation. In our approach the timing failures, which are caused by short and open resistive faults inside the gates, will affect the level of the voltage at the gate output 2. The relation between the logic propagation delay and its eventual effect on the circuit output voltage is determined by performing a switch-level analysis on CMOS primitive gates. Consequently the defected voltage, which is carrying timing disturbance, should be propagated to the primary output. As the delay size is relatively small the voltage changes do not cause logical problems at the gate output and it is therefore difficult to trace these faulty signals to the output. To resolve this problem we use a nine-valued voltage model on top of a five-valued voltage model to propagate faulty signals. The main concept is that those faults that are causing logical faults in the nine-valued

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 653–660, 2008.

voltage model are still delay faults in the five-valued voltage model. The dynamic behavior of resistive defects tends to delay the correct logic state propagation at the gate output. Generally, the delay is caused by certain fault locations with respect to the input vector the gate is subjected to, defect resistance of the fault, and the technology variation. In most cases, these faults in the five-valued voltage model only disturb the logic propagation time without adversely changing the functional output. As a result, the output voltage fluctuates between ranges of intermediate voltage value. The disturbance of the propagation time however, can greatly affect the functional output in a nine-valued voltage model. By reducing feature sizes, resistive fault occurrences are expected to rise. Hence, their effect on the logic voltage-levels in static CMOS primitive gates subject to technologies such as 65nm, 45nm and 32 nm can be determined. A description of the dynamic effect of resistive faults on propagation delay and output voltage in nanometer technology 2 is presented in Section 2. Section 3 presents a quick review of our recently published algorithm 1 for fault propagation in transistor level. Section 4 demonstrates how this algorithm is also able to propagate delay faults using two voltage models. Section 5 presents the experimental results and Section 6 concludes this paper and is followed by a list of cited references.

2 Dynamic Effects of Resistive Faults and Output Voltage

It is assumed that existing bridging fault models describe shorts between logical nodes with a short resistance of 0 Ω only. Many studies regarding the delay defect synthesis with respect to reducing transistor sizes discuss gate–level fault modeling and fault diagnosis. Generally, the voltage degradation caused by resistive physical defects is accounted for as intermediate node voltage. There has been little focus in the area of propagation of the additional delay to the circuit output. Our previous publication in 2 demonstrates how to calculate the voltage changes at gate output that cause a delay rather than a logic fault. Later in this paper this delay is propagated to the output. The switch-level delay fault described in 1 involves the simulations based on fixed capacitive load. For a precise analysis, parasitic resulting from the MOSFET are included for delay estimation due to resistive faults 2.

3 Review of Fault Propagation in Transistor Level

In our previous research 1 we introduced a novel fault synthesis algorithm for modeling CMOS circuits with an arithmetic solution for circuit verification and fault synthesis. This new approach is capable of simulating multiple fault injection into the circuit and speeds up switch-level simulation. Another advantage of this algorithm is its application in the mapping of single and multiple faults from switch-level to gate level as well as its function as a multi level model. This paper presents a unique method to propagate delayed signals created by resistive short and open faults. This method converts a circuit to a graph, finds the arithmetic equation for it and eventually synthesizes faults and generates the outputs. Some terms such as ‘Connection Node’ and ‘Circuit Graph Model’ that are used in this paper are fully described in 1.

4 Delay Pzropagation of Resistive Faults in Deep Sub-microns

Many multiple delay fault based diagnosis methods have been published56. Dastidar and Touba 5 proposed an approach for multiple delay-fault diagnosis based on static timing information. Authors in 6 investigated the effectiveness of n-detection tests to diagnose failure responses caused by multiple stuck-at and bridging faults. Our approach is capable of diagnosing multiple delay faults as well as static faults in switch level, which is more accurate and applicable even when the delay size is relatively small. This approach uses two model of IEEE Standard 1164-1993: MVL5 and MVL9 (Multi-Valued Logic System) and is able to propagate delays in MVL5 using a MVL9.

Table 1. Mapping MVL9 and MVL5

Nine-valued voltage model	Five-valued voltage model
1	1
H	
X	
X	H
U	U
W	L
W	0
L	
0	

It is difficult to propagate any fault in a digital circuit when there are no logical changes in signals. Most algorithms are able to propagate timing disturbance as soon as there is a logical failure in circuit functionality, but in the absence of logical failure these methods are useless. To solve this problem we use a MVL9 on top of a MVL5. When there is a slow-to-rise or slow-to-fall delay because of a resistive open or short fault in the circuit the output voltage of the gate will be affected accordingly. As explained in Section 2, this change is not big enough to cause a logic change in MVL5 but may change the logic in MVL9. Table 1 shows when logic 0 or 1 is considered for MVL5 logic 0, L, W or 1, H, X can be considered for MVL9 for lower or upper boundaries. The rest of this section describes how to convert a circuit to an arithmetic equation using its graph and to propagate a signal in MVL9 as a logic failure.

4.1 Arithmetic Equation of the Circuit

The algorithm uses the behavior of CMOS transistors in digital circuits 3 and describes the circuit in an arithmetic equation 1.

An equation describes circuit behavior in detail according to all possible input combinations. In this arithmetic model each transistor is considered as a function such as ‘P’ and ‘N’. Each function has two arguments as inputs and a returning logical value that is considered for Drain. P (G, S) and N (G, S) are the syntaxes for the functions. The ‘P’ function is used for P-type transistors and the ‘N’ function is used for N-type transistors. The first argument or ‘G’ is the value of Gate and the second argument or ‘S’ is the value of Source for each transistor. The result of the function will be calculated with the logic value at the drain. The value of each function can be

Table 2. CMOS behavior lookup table, Nine-valued voltage model

Gate	Source	Drain P (G, S)	Drain N (G, S)	Gate	Source	Drain P (G, S)	Drain N (G, S)
L	L	W	Z	H	L	Z	W
L	H	X	Z	H	H	Z	X
L	Z	Z	Z	H	Z	Z	Z
L	1	H	Z	H	1	Z	H
L	0	L	Z	H	0	Z	L
L	U	U	Z	H	U	Z	U
L	X	X	Z	H	X	Z	X
L	W	W	Z	H	W	Z	W
Z	L	U	U	1	L	Z	L
Z	H	U	U	1	H	Z	H
Z	Z	U	U	1	Z	Z	Z
Z	1	U	U	1	1	Z	1
Z	0	U	U	1	0	Z	0
Z	U	U	U	1	U	Z	U
Z	X	U	U	1	X	Z	X
Z	W	U	U	1	W	Z	W
0	L	L	Z	U	L	U	U
0	H	H	Z	U	H	U	U
0	Z	Z	Z	U	Z	U	U
0	1	1	Z	U	1	U	U
0	0	0	Z	U	0	U	U
0	U	U	Z	U	U	U	U
0	X	X	Z	U	X	U	U
0	W	W	Z	U	W	U	U
X	L	W	W	W	L	W	X
X	H	W	X	W	H	X	X
X	Z	W	Z	W	Z	Z	X
X	1	W	H	W	1	H	X
X	0	W	L	W	0	L	X
X	U	W	U	W	U	U	X
X	X	W	X	W	X	X	X
X	W	W	W	W	W	W	X

U Uninitialised, X Forcing Unknown, 0 Forcing 0, 1 Forcing 1, Z High Impedance, W Weak Unknown, L Weak 0, H Weak 1.

derived from a lookup table (Table 2) in MVL9. For example, if given the logical values G = L and S = 1 then, according to table 2, these functions will be P (L, 1) = 1 and N (L, 1) = Z.

Equation (1) is an arithmetic evaluation for NOT gate.

$$Y = P\,(G, S)\ \nabla\ N\,(G, S) \tag{1}$$

In NOT gate the drain of a P-channel transistor is connected to the drain of an N-channel transistor, each with its own logical value. The outcome is a dominant value for this connection. For instance, if the drain of the P-channel transistor has the value '1' and the drain of N-channel transistor has the value 'Z' then the dominant value '1' is considered for this node and symbolized as '∇'. Table 3 shows the dominant logic value for connection node in MVL9.

Table 3. Connection node equivalent value lookup table, Nine-valued V

∇	U	X	0	1	Z	W	L	H	-
U	U	U	U	U	U	U	U	U	U
X	U	X	X	X	X	X	X	X	X
0	U	X	0	X	0	0	0	0	X
1	U	X	X	1	1	1	1	1	X
Z	U	X	0	1	Z	W	L	H	X
W	U	X	0	1	W	W	W	W	X
L	U	X	0	1	L	W	L	W	X
H	U	X	0	1	H	W	W	H	X
-	U	X	X	X	X	X	X	X	X

4.2 Delay Propagation

The actual delay, which is the result of a resistive short or open fault in a digital circuit, will affect voltage and this voltage must pass through a path to the output. To illustrate how this algorithm works note the following examples.

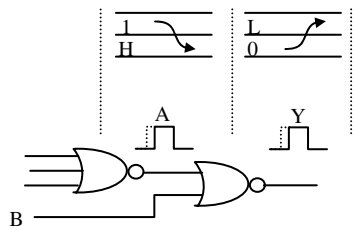


Fig. 1. Logic change in MVL9

Fig.1. shows a three-input NOR gate connected to a two-input NOR gate. Assume there is a delay at the output of three-input NOR gate caused by a 5 KΩ resistor between drain D3 and VDD. According to table 4 (Section 5) for 180 nm technology, output voltage of slow-to-rise is 1.16 V for a delay of 0.37 µs. Further explanation will be given in the Experimental Result section. This value is considered as 'H' signal in MVL9 instead of '1'. Now this signal must pass through a two-input 'NOR' gate using the arithmetic equation of the circuit. Fig.2. (a) shows the transistor level of two-input NOR gate and (b) shows the arithmetic model of the circuit. As explained earlier, output of the circuit 'Y' can be calculated by equation (2) as follows:

$$Y = P2\ (G, P1(G, S))\ \nabla\ (N1(G, S)\ \nabla\ N2(G, S)) \quad (2)$$

Equation (3) represents equation (2) after replacing actual signals for gate 'G' and source 'S'.

$$Y = P2\ (B, P1(A, 1))\ \nabla\ (N1(A, 0)\ \nabla\ N2(B, 0)) \quad (3)$$

The delayed signal 'H', which is caused by a resistive short fault in a three-input NOR gate is considered as input signal 'A' in the above equation. To propagate this signal to the 'Y' output input 'B' must have the value of '0'. The result is shown in equation (4).

$$Y = P2\ (0, P1(H, 1)) \nabla (N1(H, 0) \nabla N2(0, 0)) \quad (4)$$

Values of function 'P' and 'N' can be derived from lookup table 2. The value of connection node '∇' must be taken from lookup table 3 and the final value for 'Y' will be equal to 'L'. All the steps are shown as follows:

P1 (H, 1) = Z

N1 (H, 0) ∇ N2 (0, 0) = L ∇ Z = L

Y = P2 (0, Z) ∇ L = Z ∇ L = L

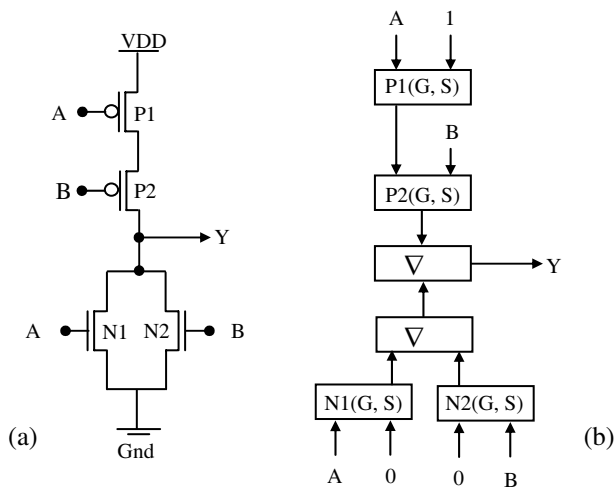


Fig. 2. (a) Transistor level of two-input NOR gate and (b) Arithmetic model

As shown in Fig.1. when the logic value for signal A in MVL9 changes from '1' to 'H' as a result of the resistive fault in the circuit, the logic value for output 'Y' will change from '0' to 'L'. This change at output represents a delay of 0.37 μs. at the same time in MVL5 model signal 'A' and 'Y' remaining the same value '1' and '0' regardless of presenting a delay fault in the circuit. This is shown in table 1 while mapping two models. As previously mentioned this arithmetic algorithm can be applied for the circuit with multiple delays and may affect each other at the output. One delay may also compensate for another and vanish through several stages. Sometimes two or more delays may combine and cause a logical failure in MVL5.

5 Experimental Results

Experimental results were obtained in two phases. In the first phase the functionality of the algorithm with the Cadence Spectra simulator was checked. The experiment was completed for all CMOS gates at transistor level. In the second phase, Delay Fault Simulation (DFS) software was developed and run on several ISCAS85 and ISCAS89 benchmark circuits. This software is able to convert a whole circuit to a graph using the developed algorithm, and then convert this graph to an arithmetic

model. The arithmetic model is able to simulate all kinds of faults, but in this paper we narrowed our test and results on delay faults for 180 nm technology. For clarification, Table 4 lists all delays and related output voltage for a 3-input NOR gate with resistive faults between drain D_1 through D_3 and V_{DD} in 180 nm technology. Table 4 shows that delay size t_{PLH} and similarly voltage output V_{OUT} for different resistor size RS are almost the same except for 1k in fault sites closer to power like D_3 which can be considered a short circuit. As some fault injections for 1k may cause logic problems as shown in Table 4 for D_3 to V_{DD} (Logic 1 $\geq$ 1.62) and it is not going to be argued in our research area for delay faults we ignored 1k in our experimental result.

Table 4. 180 nm - V_{OUT} v/s t_{PLH}, $I_1I_2I_3$ = 000, t_{PLH} in µs, V_{OUT} in volt

	D3 to VDD		D2 to VDD		D1 to VDD	
RS (Ω)	tPLH	VOUT	tPLH	VOUT	tPLH	VOUT
15K	0.45	1.07	0.47	1.04	0.49	1.02
10K	0.42	1.09	0.46	1.05	0.49	1.03
5K	0.37	1.16	0.44	1.08	0.48	1.03
1K	0.18	1.69	0.32	1.24	0.44	1.08

To apply MVL9 in simulation software, the voltage range between V_{DD} and Ground is divided in intervals representing one of the logic levels at MVL9. Eventually the voltage values generated based on the injected fault and input values (0) will be assigned to a logic value in that interval. For instance, the logic value for fault D_3 to V_{DD} in Table 4 will be considered as X (forcing unknown) in MVL9. Finally pseudo-random input test sets were obtained from gate level test generation tools to complete this simulation. As expected, results indicate that gate level test vector sets detect less switch level faults. Switch level fault coverage was less than gate level fault coverage. This result, while not surprising, confirms the fact that switch fault simulation can be a better design verification tool, as a larger test set would be required to achieve switch level fault coverage similar to the gate level fault coverage. Any possible switch level faults that are undetected for the gate level test set may be detected by the larger switch level test set. Fault Coverage (FC) simulation results for resistive short delay faults are shown in Table 5. Test simulation results can be achieved for different technologies.

Table 5. Fault coverage Simulation results for resistive short delay fault

Circuit	# of Switches	# of Faults	Delay Propagation (%FC)
C17	24	30	100
C2670	6212	1247	64
C7552	18802	2667	67
S27	66	70	99
S298	582	463	67
S1238	2662	1054	58
S13207	30984	4705	52
S38417	85912	9816	48

6 Summary and Conclusion

This paper proposes a novel algorithm for diagnosing and propagating gate-delay faults in deep sub-micron in two phases:

Phase one is designed to convert a circuit with a resistive model as explained in 2. It calculates the accurate time and voltage disturbance at gates output for all existing technologies i.e. 65nm, 45nm and 32 nm. Our experiment is based on 180 nm technologies.

Phase two propagates the gate-delay fault to the primary output using a powerful arithmetic model of the circuit in transistor level. To propagate delay faults even with a relatively small delay size the algorithm maps a MVL9 on top of a MVL5 in order to propagate those delays that do not cause logical failures. To implement these two phases we used Cadence tools for accurate testing. Delay Fault Simulation (DFS) software was also developed to inject faults into the circuit and measure the fault coverage. Recent research in 4 describe physically realistic resistive open and bridge fault models incorporating both functional and delay effects of spot defects. However, there is no method to estimate the fault coverage. In our approach a switch-level algorithm for delay faults has been presented and its applicability in robust delay fault testing was demonstrated. This algorithm would be especially useful for varied applications such as switch level min-max mode grading of robust/non-robust delay test vectors and analysis of dynamic hazards for delay fault diagnosability for general switch-level circuits. We intend to explore these applications in the future.

References

1. Javaheri, M.R., Sedaghat, R., Kant, L., Zalev, J.: Verification and Fault Synthesis Algorithm at Switch-Level. Journal of Microprocessors and Microsystems (2005)
2. Sedaghat, R., Kunchwar, M., Abedi, R., Javaheri, R.: Transistor-level to Gate-level Comprehensive Fault Synthesis for n Input Primitive Gates. Journal of Microelectronics Reliability (2006)
3. Kunchwar, M., Sedaghat, R.: Dynamic Behavior of Resistive Faults in Nanometer Technology. International Journal of Microelectronics Reliability, 1st revision (2006)
4. Alt, J., Mahlstedt, U.: Simulation of non-classical faults on the gate level – fault modeling – Institute fur Theoretische Elektrotechnik. In: 11th VLSI Test Symposium, Universität Hannover, Germany, pp. 351–354 (April 1993)
5. Li, Z., Lu, X., Qiu, W., Shi, W.: Walker, A circuit level fault model for resistive opens and bridges D.M.H. In: VLSI Test Symposium, 2003. Proceedings. 21st 27 April-1, pp. 379–384 (May 2003)
6. Dastidar, J.G., Touba, N.A.: A Systematic Approach for Diagnosing Multiple Delay Faults. In: Proc. of the IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, pp. 211–216 (1998)
7. Wang, Z., Tsai, K.-H., Marek-Sadowska, M., Rajski, J.: An Efficient and Effective Methodology on the Multiple Fault Diagnosis. In: Proc. of the International Test Conference, pp. 329–338 (2003)

On the Importance of the Number of Fanouts to Prevent the Glitches in DPA-Resistant Devices*

Amir Moradi[1], Mahmoud Salmasizadeh[2], and Mohammad Taghi Manzuri Shalmani[1]

[1] Department of Computer Engineering, Sharif University of Technology, Tehran, Iran
[2] Electronic Research Center, Sharif University of Technology, Tehran, Iran
a_moradi@ce.sharif.edu, {salmasi,manzuri}@sharif.edu

Abstract. During the last years several logic styles have been proposed to counteract power analysis attacks. This article starts with a brief review of four different logic styles namely RSL, MDLP, DRSL, and TDPL. This discussion continues to examine the effect of the number of fanouts in power consumption of a CMOS inverter. Moreover, it is shown that insertion of delay elements in typical CMOS circuits is not adequate to prevent the glitches and information leakage unless the fanouts of input signals are balanced. Whereas enable signals have to be classified according to the depth of combinational circuits implemented using pre-charge logic styles, we show that the number of fanouts of enable signals (which is equal to the number of gates in each depth) is a significant factor in determining the interval between arrival time of the consecutive enable signals.

Keywords: DPA, Glitches, Delay Elements, Fanouts, Dual-Rail Logics, Pre-Charge Logics.

1 Introduction

Since power analysis attack was introduced by P.Kocher *et al.* in 1999 [4], security of implementations has been considered as a new field of study by several researchers. Nowadays information leakage of cryptographic devices plays a significant role in security evaluation of implementations. Hamming weight of the modified data in a register was the first hypothetical power consumption model in power analysis attacks whose aim was to exploit the secret information of a CMOS implementation. Then, some methods were proposed to resist Simple Power Analysis (SPA) and Differential Power Analysis (DPA) attacks. Masking techniques are the most popular suggested way to protect an implementation against DPA. Some of masking methods have been designed in the algorithm level by the addition of a random value to the input or intermediate values [7]; the other ones have used a random bit for each signal or a batch of signals at the gate level [3].

Mangard *et al.* have presented a more accurate hypothetical model for the power consumption of combinational circuits [5]. They used the number of transitions which

* This project is partially supported by Iran National Science Foundation and Iran Telecommunication Research Center.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 661–670, 2008.

occur in the combinational circuit during the change of inputs instead of Hamming weight of the values changed in registers. Mangard *et al.* have revealed that the glitches are caused by the XOR gates of the masked multipliers in a masked AES Sbox [6]. They have proposed two methods to prevent glitches: (i) the insertion of delay elements to balance the arrival time of all input signals of the XOR gates and (ii) the usage of enable signals for each depth of the XOR gates.

Pre-charge logics such as Sense Amplifier Based Logic (SABL) [11], Wave Dynamic Differential Logic (WDDL) [12], and Three-Phase Dual-Rail Pre-Charge Logic (TDPL) [1] have been proposed to counter power analysis attacks. Additionally, Masked Dual-Rail Pre-Charge Logic (MDPL) [8], which has mixed the masking technique at the gate level and pre-charge logics, has been proposed to prevent information leakage. Recently, it has been shown that MDPL leaks information too when there is difference between the delay time of input signals [9]. Each input signal is the output of a gate in the previous level, but all input signals are not the output of the gates at the same depth; consequently, the arrival time of the transitions in input signals may differ.

In this article, we show that the insertion of delay elements to balance the delay time of input signals is not sufficient to prevent glitches, *i.e.* there is information leakage even all input signals of each gate are the output of the gates at the same depth. Obviously, the major factor of the power consumption of a logic gate relates the required current to charge and discharge the capacitive load of the gate output. In contrast, the number of fanouts of a gate determines its output capacitance. Thus, the number of fanouts plays a significant role in power consumption of logic gates, and the insertion of delay elements is essential to prevent glitches but not sufficient. Therefore, the fanout of all input signals should be balanced after the insertion of dummy gates. Moreover, we show that the glitches occur in some DPA-resistsnt logic styles unless the interval between the arrival time of pre-charge signals is specified on the base of their fanouts.

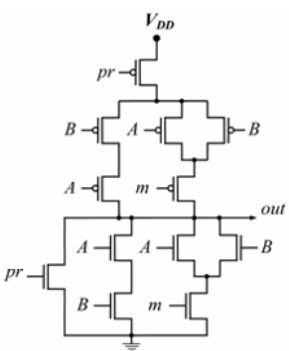


Fig. 1. Typical RSL NAND gate

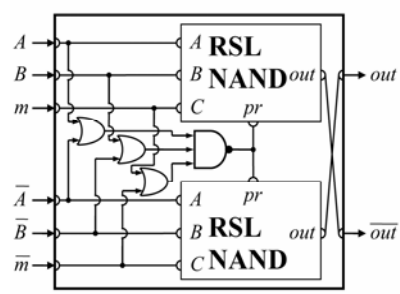


Fig. 2. Schematic of DRSL AND gate

Section 2 of this article briefly reviews four different logic styles. Some aspects of the power consumption of a CMOS gate and the effect of fanout enlargement in power consumption are illustrated in Section 3. Also, the simulation results of a

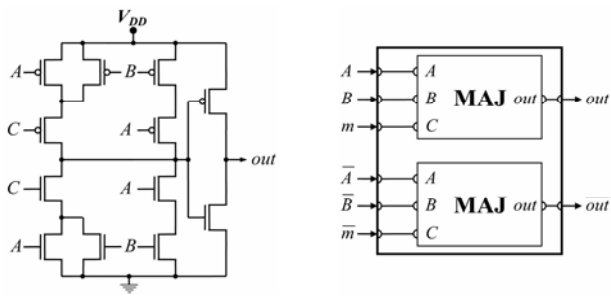
typical fully combinational circuit for each logic style are shown in Section 4. In this Section the effect of the number of fanouts in alternative logic styles is examined. Finally, conclusions are formulated in Section 5.

2 Various Logic Styles

2.1 RSL

Random Switching Logic (RSL) has been presented based on the combination of the masked logic gates and single-rail pre-charge logic [10]. All inputs of an RSL gate are synchronized by an enable signal. Figure 1 shows the typical RSL 2-input NAND gate. An RSL NOR gate is constructed by complementing the mask value, $\overline{m}$.

In fact, each RSL gate contains a pre-charge signal, *pr*, and it does not evaluate the output unless *pr* goes LO. Consider a combinational circuit which is implemented using RSL; all input signals can be changed while *pr*s stay at HI. Then, *pr* signals are changed to LO consecutively to prevent the glitches. Also, a mask bit, *m*, is used for all gates to randomize the intermediate values. Thus, the circuit is glitch free and its transitions depend on the value of the mask bit.



(a) (b)

Fig. 3. (a) Schematic of a CMOS majority gate and (b) an MDPL AND gate

2.2 MDPL

A good mixture between masking at the gate level and complementary pre-charge logic has constructed Masked Dual-Rail Pre-Charge Logic (MDPL) [8]. Figure 3.a shows the schematic of a CMOS majority gate which is used to construct MDPL gates. Figure 3.b shows how an MDPL AND gate can be constructed using majority gates. It is well known that the inverted gates are built easily by swapping the complementary output wires. Additionally, the MDPL OR gate is constructed with swapping *m* and $\overline{m}$ signals in the MDPL AND gate. Although MDPL AND/NAND and OR/NOR gates are built with a little difference, XOR/XNOR gate is built using three MDPL AND/NAND gates because XOR is not a monotonic function [8].

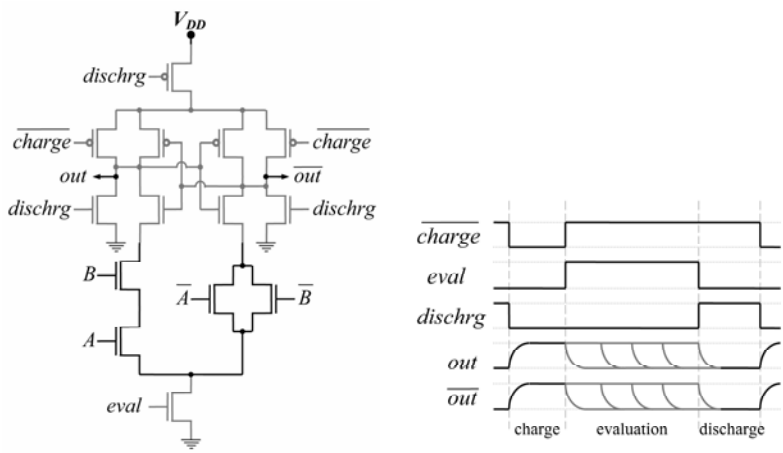
MDPL gates do not have any pre-charge or enable signal; the structure of majority gates helps preventing glitches. Thus, the difference between routing capacitance of

complementary signals may cause information leakage. However, the usage of complementary mask bits leads to avoid the dependency of power consumption on input values. This makes MDPL perfectly suitable for semi-custom designs.

2.3 DRSL

Dual-Rail Random Switching Logic (DRSL) is complementary form of RSL, but the pre-charge signal is constructed internally, and it does not need an external module to schedule the pre-charge signals [2]. The main idea of DRSL is to eliminate the control unit which makes the pre-charge signals for each logic gate. The schematic of an DRSL AND gate is shown in Figure 2. A simple control box has been added in each gate separately to prepare the needed transition in pre-charge signal.

When at least one input pair is at pre-charge phase, *e.g.* $(a, \overline{a})$ equals (LO, LO), the output of the corresponding OR gate (of control box) is LO. As a result, the output of the NAND gate, pre-charge signal, is set to HI and two internal RSL NAND gates are in pre-charge phase unless all complementary inputs are in evaluation phase. Other DRSL gates are constructed similar to the structure of MDPL gates.



(a) (b)

Fig. 4. (a) Schematic and (b) timing diagram of DRSL AND gate

2.4 TDPL

Three control pre-charge/discharge signals have been used to build a new logic style, which guarantees that two transitions occur at each output signal in every evaluation cycle. It is called Three-Phase Dual-Rail Pre-Charge Logic (TDPL) [1]. Figure 4.a illustrates typical TDPL NAND/AND gate; the gray elements, five nmos and five pmos transistors, are fixed in all TDPL gates, and the black elements, all in nmos network, construct the desired logical gates. As shown in Figure 4.b, there are three phases to generate a valid output. Signal $\overline{charge}$ makes a high transition at all output

signals then the correct output values are evaluated, and discharge phase guarantees high to low transitions. Consequently, even if the capacitive loads of complementary signals differ, the power consumption for a complete charge-evaluation-discharge period is constant. However, it needs a much more complex control module to schedule *eval*, *discharge*, and $\overline{charge}$ signals.

3 Fanouts and Glitches in CMOS Gates

Equations (1) to (4) show the power consumption specification of CMOS circuits [13]. P_{Static} depends on the number of gates, the leakage current of the used technology, and etc. $P_{Short\text{-}Circuit}$ depends on the activity of the circuit, *i.e.* the number of transitions which occur during a period of time in the gates' output. P_{Switch} relates to the required current for charging or discharging the load capacitance, C_L.

$$P_{Total} = P_{Static} + P_{Short\text{-}Circuit} + P_{Switch} \tag{1}$$

$$P_{Static} = \sum LeakageCurrent \times SupplyVoltage \tag{2}$$

$$P_{Short\text{-}Circuit} = \frac{\beta \cdot (V_{DD} - 2 \cdot V_t)^3}{12} \cdot f \tag{3}$$

$$P_{Switch} = C_L \cdot {V_{DD}}^2 \cdot f \tag{4}$$

Suppose that we want to analyze the power consumption characteristics of a CMOS inverter when only once does its input rise and fall. $P_{Short\text{-}Circuit}$ is the same for each transition and it depends only on the rise and fall time of the input signal, but P_{Switch} depends on the C_L which is determined by the input capacitance of the next gate. Obviously, when the fanout of a gate is increased, the number of capacitors which are connected to its output becomes more.

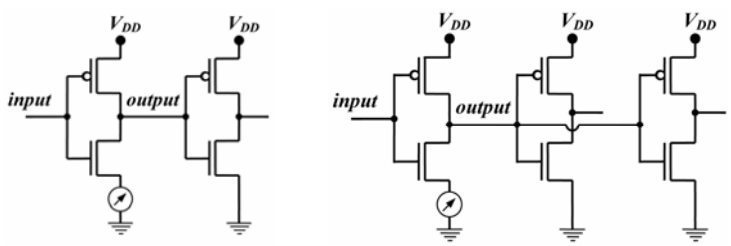


Fig. 5. Two test circuits in order to examine the effect of fanouts on power consumption of a typical CMOS inverter

Figure 5 shows two different test circuits in which the number of fanouts of the considered NOT gate is different. Figure 6 presents a comparison between the simulation results of each circuit under the same condition for a rise and a fall on the input

signal. All simulations have been done by HSPICE using TSMC 0.18μm library and 1.8v as V_{DD} supply. Each transistor is designed with a width W=1μm and 3μm in nmos and pmos networks respectively, and the minimum gate length, L=0.18μm, is assumed for all.

As shown in Figure 6, the rise and fall time of the output signal becomes more by increasing the number of fanouts. Furthermore, the consumed energy of the fall and rise of the output signal has been increased, but a little difference between the peaks of the passed current is observed.

Power sampling in real world is done by measuring the current which is passed through V_{SS} or V_{dd} net. Current is measured by sampling the differential voltage of a small resistor (typically 1Ω) in V_{SS} or V_{dd} net. Several factors affect the accuracy of the measurement; besides, the sampling resolution is not comparable with the simulation results. Accordingly, if the peaks of the sampled current values, *i.e.* an index of the instantaneous power consumption, are used in DPA attacks, the effect of the number of fanouts is negligible. In contrast, if the sum of values measured in a period of time, *i.e.* the integral of the passed current equals a coefficient of the consumed power, is used, the number of fanouts of each signal is a significant parameter.

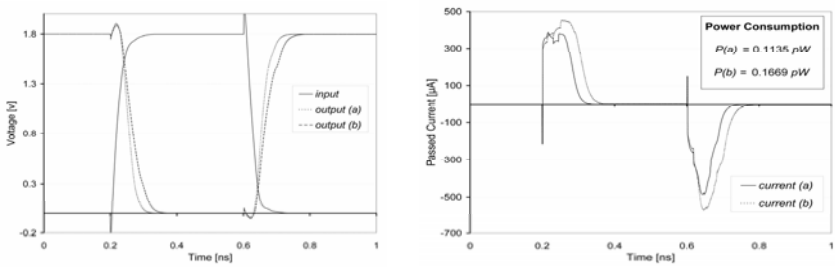
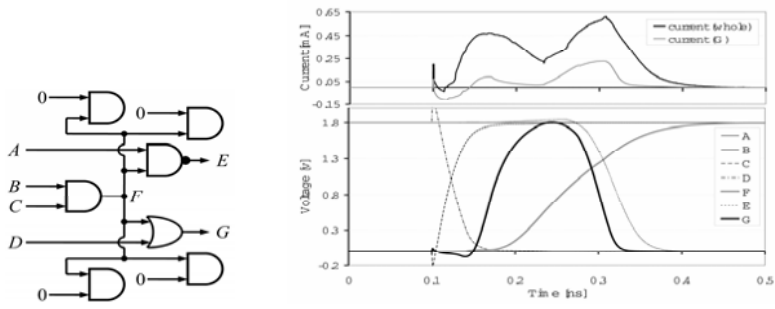


Fig. 6. Simulation results of two circuits presented in Figure 5

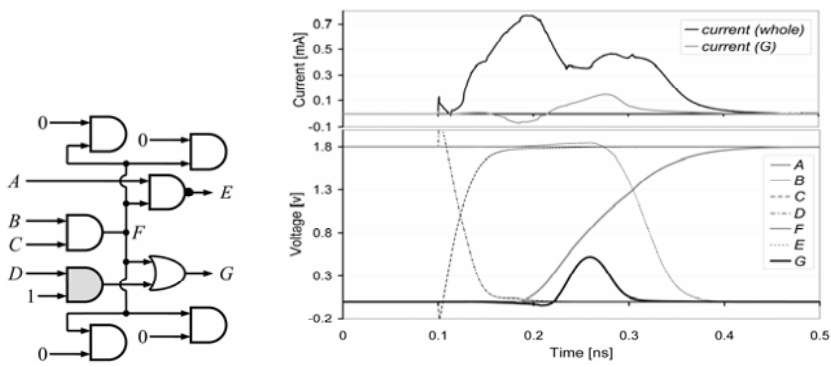


(a) (b)

Fig. 7. (a) test circuit and (b) simulation results of the test circuit implemented using CMOS standard logic cells when (A,B,C,D) changed from (1,1,0,1) to (1,1,1,0)

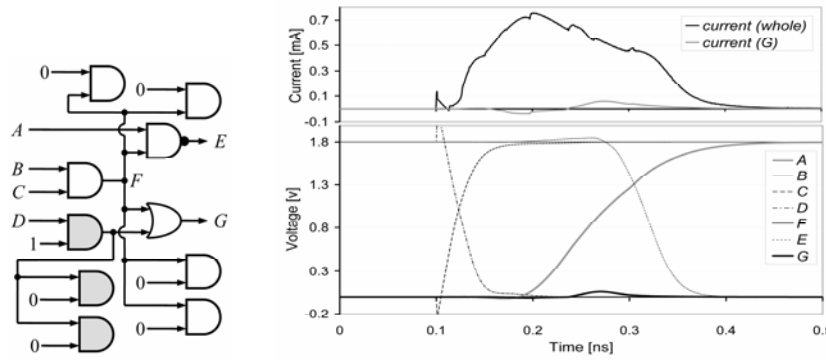
A test circuit, shown in Figure 7.a, is considered to show the glitches and the effect of delay elements and fanouts. Suppose (A, B, C, D), which are the inputs of the test

circuit, change from (1,1,0,1) to (1,1,1,0); thus, the output signal, *E*, changes from 1 to 0, *F* changes from 0 to 1, and *G* stays at 0 probably. Figure 7.b shows the simulation results of the transitions in the implemented test circuit using CMOS standard logic cells. A little glitch occurs on signal *G*, and it influences the current of power supply.



(a) (b)

Fig. 8. (a) The test circuit which is delayed at signal D, (b) The simulation results with standard CMOS logic gates



(a) (b)

Fig. 9. (a) The delayed and balanced test circuit, (b) simulation results with standard CMOS logic gates

According to Figure 8.a, a dummy AND gate has been inserted at net *D*; the AND gate has been selected to equalize the delay time of signals *D* and *F*, which are the input signals of the gate *G*. If the NOR gate, *G*, had more inputs, all of them would be delayed and be synchronized according to the last one. The simulation result of the delayed circuit is presented in Figure 8.b. Still, a little glitch can be observed at the

output of the gate *G* because the capacitive load of the signal *F* is much more than the delayed signal. Thus, the delay time of the signal *F* is still more than the delayed signal. Additionally, the passed current of the power supply and the amount of the consumed power that relates to the gate *G* are noticeable. Consequently, as shown in Figure 9.a, two other dummy gates have been added to increase the load capacitance of the delayed signal. Figure 9.b, which exhibits the simulation results of the circuit in part (a), shows an improvement to prevent the glitch at signal *G*. The passed current and the consumed power are shown that the small peak which occurs at the output of the gate *G* is negligible. In addition, it is possible to restrain the small peak by balancing the number of fanouts exactly, means four dummy gates are inserted to match the capacitive loads.

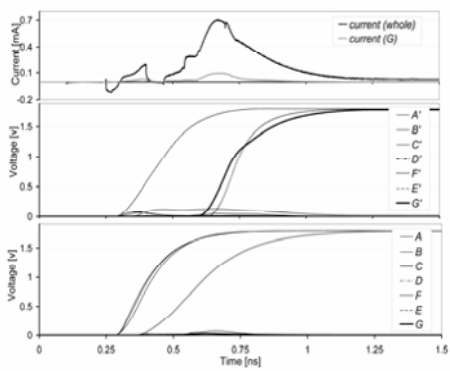


Fig. 10. Simulation results of the test circuit which is implemented with MDPL gates

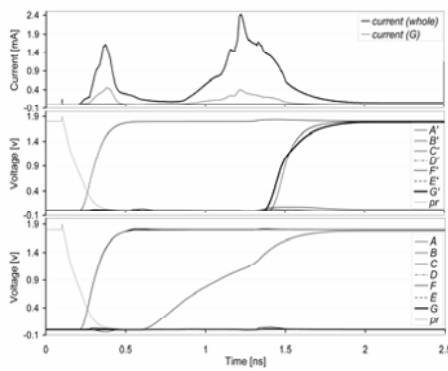


Fig. 11. Simulation results of the test circuit which is implemented with DRSL gates

4 Fanouts and Glitches in Other Logic Styles

As mentioned previously, the majority gates in MDPL construct a glitch free circuit. Also, MDPL circuits contain no pre-charge or enable signal to control the evaluation phase. Figure 10 shows the simulation result of the test circuit presented in Figure 7.a that is implemented in MDPL style. Where m and $\overline{m}$ were considered as 0 and 1 respectively. Obviously, no glitches occur on every logic signal. Furthermore, the number of fanouts does not play any role in the presence of glitches, but it may affect on the circuit delay time.

Also, the test circuit was implemented with DRSL elements. Clearly, it does not need an external module to schedule pre-charge signals; the pre-charge signal is controlled by an internal module inside each DRSL gate. Figure 11 shows the simulation results of the DRSL test circuit. m and $\overline{m}$ are considered as 0 and 1 respectively. Never do the results show any glitches on the output signals. However, its delay time is much more than the corresponding MDPL circuit because of the delay time of pre-charge control modules. Obviously, the number of fanouts does not affect on the presence of glitches in this logic style.

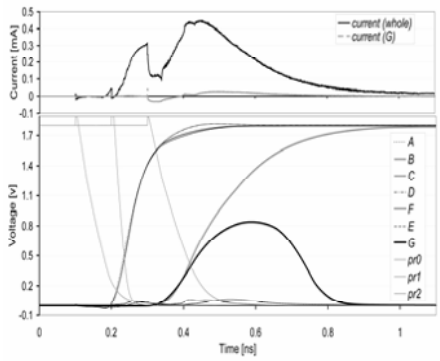


Fig. 12. Simulation results of the test circuit implemented in RSL style

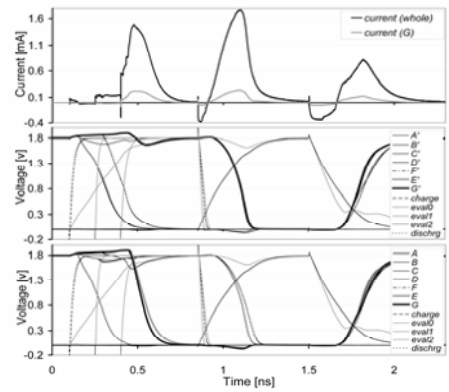


Fig. 13. Simulation results of the test circuit implemented using TDPL gates

In contrast, the circuits that are implemented using RSL elements require a control module to handle the pre-charge signals. The most important factor in this module is the interval between the arrival times of consecutive pre-charge signals. Figure 12 shows the simulation results of the test circuit that is implemented in RSL style. Three pre-charge signals, *i.e. pr0* to *pr2*, are used; 100*p*s was considered as the time interval between two respective *pr*s. This interval is sufficient for *pr1*, and signal *F* is glitch free. However, 100*p*s is short for the difference between *pr2* and *pr1*, and a little glitch is observable at signal *G*. The time interval between *pr2* and *pr1* must be greater than the fall or rise time of *F*. Thus, the number of fanouts of gate *F* which affects on the rise/fall time is a significant factor to prevent glitches.

In TDPL circuits all of $\overline{charge}$ and all of *discharge* signals are connected together, but the difference between arrival time of *eval* signals prevents the glitches. Similarly, the time interval between respective *eval*s must be specified accurately, and the number of fanouts plays a significant role on the time adjustments. Figure 13 shows the simulation result of the test circuit that is implemented in TDPL style. We specified time intervals precisely, and no glitch is observed on logic signals.

5 Conclusions

In the first part of this paper, structure of four logic styles (RSL, MDPL, DRSL, and TDPL), which have recently been proposed to counteract power analysis attacks, were reviewed briefly. Then, we have presented the effect of the number of fanouts of a simple CMOS inverter on the power consumption and on the peak of the passed current. Also, the effects of the delay element insertion and balancing the number of fanouts on standard CMOS implementations were illustrated. Simulation results show that the insertion of dummy gates is not sufficient to prevent the glitches unless the number of fanouts of all input signals are balanced for each gate separately.

At the second part, the simulation results of a typical test circuit which is implemented using the considered logic styles were presented. Clearly, MDPL and DRSL

implementations are completely glitch free because the MDPL uses CMOS majority gates to construct complementary logic gates and DRSL builds the pre-charge signals internally without the usage of an external scheduling module. However, RSL and TDPL implementations need the time interval between pre-charge/evaluation signals to be specified accurately on the base of the delay time of each logic gates. On the other hand, the number of fanouts affects the fall and rise times significantly. Thus the number of fanouts is an important factor in the time adjustments also in preventing the glitches.

References

1. Bucci, M., Giancane, L., Luzzi, R., Trifiletti, A.: Three-Phase Dual-Rail Pre-charge Logic. In: Goubin, L., Matsui, M. (eds.) CHES 2006. LNCS, vol. 4249, pp. 232–241. Springer, Heidelberg (2006)
2. Chen, Z., Zhou, Y.: Dual-Rail Random Switching Logic: A Countermeasure to Reduce Side Channel Leakage. In: Goubin, L., Matsui, M. (eds.) CHES 2006. LNCS, vol. 4249, pp. 242–254. Springer, Heidelberg (2006)
3. Fischer, W., Gammel, B.M.: Masking at Gate Level in the Presence of Glitches. In: Rao, J.R., Sunar, B. (eds.) CHES 2005. LNCS, vol. 3659, pp. 187–200. Springer, Heidelberg (2005)
4. Kocher, P.C., Jaffe, J., Jun, B.: Differential Power Analysis. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 388–397. Springer, Heidelberg (1999)
5. Mangard, S., Pramstaller, N., Oswald, E.: Successfully Attacking Masked AES Hardware Implementations. In: Rao, J.R., Sunar, B. (eds.) CHES 2005. LNCS, vol. 3659, pp. 157–171. Springer, Heidelberg (2005)
6. Mangard, S., Schramm, K.: Pinpointing the Side-Channel Leakage of Masked AES Hardware Implementations. In: Goubin, L., Matsui, M. (eds.) CHES 2006. LNCS, vol. 4249, pp. 76–90. Springer, Heidelberg (2006)
7. Oswald, E., Mangard, S., Pramstaller, N., Rijmen, V.: A Side-Channel Analysis Resistant Description of the AES S-box. In: Gilbert, H., Handschuh, H. (eds.) FSE 2005. LNCS, vol. 3557, pp. 413–423. Springer, Heidelberg (2005)
8. Popp, T., Mangard, S.: Masked Dual-Rail Pre-Charge Logic: DPA Resistance without Routing Constraints. In: Rao, J.R., Sunar, B. (eds.) CHES 2005. LNCS, vol. 3659, pp. 172–186. Springer, Heidelberg (2005)
9. Suzuki, D., Saeki, M.: Security Evaluation of DPA Countermeasures Using Dual-Rail Pre-charge Logic Style. In: Goubin, L., Matsui, M. (eds.) CHES 2006. LNCS, vol. 4249, pp. 255–269. Springer, Heidelberg (2006)
10. Suzuki, D., Saeki, M., Ichikawa, T.: Random Switching Logic: A Countermeasure against DPA based on Transition Probability. Cryptology ePrint Archive, Report 2004/346 (2004), http://eprint.iacr.org/
11. Tiri, K., Akmal, M., Verbauwhede, I.: A Dynamic and Differential CMOS Logic with Signal Independent Power Consumption to Withstand Differential Power Analysis on Smart Cards. In: European Solid-State Circuits Confer., pp. 403–406 (2002)
12. Tiri, K., Verbauwhede, I.: A Logic Level Design Methodology for a Secure DPA Resistant ASIC or FPGA Implementation. In: DATE 2004, pp. 246–251. IEEE Computer Society, Los Alamitos (2004)
13. Weste, N.H.E., Eshraghian, K.: Principles of CMOS VLSI Design – A Systems Perspective, 2nd edn. Addison-Wesley, Reading (1993)

Performance Enhancement of Asynchronous Circuits

Somaye Raoufifard, Behnam Ghavami, Mehrdad Najibi, and Hossein Pedram

Computer Engineering Department, Amirkabir University of Technology
Tehran, Iran
{raoufifard,ghavamib}@aut.ac.ir, {najibi,pedram}@ce.aut.ac.ir

Abstract. This paper proposes a methodology for automatic slack matching of QDI circuits utilized in a framework of asynchronous synthesis toolset. Slack matching is the problem of adding buffers to an asynchronous pipeline design to prevent stalls and improve performance. This technique is based on Simulated Annealing method and exploits the advantages of both static and dynamic performance analysis to provide enough results in an acceptable time. The utilized performance model is a Timed Petri Net (TPN) which can extended to support choice places to capture the conditional behavior of the system. We implemented this method in the framework of asynchronous synthesis tool and optimized circuits using this technique during synthesis process. The results demonstrate that this algorithm is computationally feasible for moderately sized models. Experimental results on a large set of ISCAS benchmarks indicate that our proposed technique can achieve on average 38% enhancement for performance with 26% area penalty.

Keywords: Asynchronous Circuit, Performance Optimization, Slack Matching, PetriNet, Simulated Annealing.

1 Introduction

Stalls that are caused by miss-matches in communication rates are a major performance obstacle in asynchronous circuits. If the rate of data production is faster than the rate of consumption, the resulting design performs slower than the case where the communication rate is matched. This can be remedied by inserting pipeline buffers to temporarily hold data and allowing the producer to proceed if the consumer is not ready to accept data. This type of modification is a well-known optimization technique referred to as slack matching. This technique reduces a system's cycle time by inserting buffers into communication channels. Our goal is to present a method for automatic insertion of slack matching buffers in order to improve performance. The optimal solution for the slack matching problem has been shown to be NP-complete[2].Recent investigations cast the slack matching problem as an integer programming problem and use generic IP solvers to get a target cycle time[7],[10]. Alternative work[12] uses leveraging protocol knowledge in slack matching and present a heuristic that uses knowledge of the communication protocol to explicitly model these bottlenecks. They are an iterative algorithm to remove these bottlenecks by inserting buffers. Modeling in most of these techniques is complex,

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 671–678, 2008.

and sometimes appear to be unsolvable for large circuits. In addition, there are some other problems for exploiting these techniques. Such as handling hierarchical circuit models and modeling systems with choice. Because of these restrictions, it is difficult to apply these techniques for optimization of real asynchronous systems.

This paper presents a new method for optimizing pipelined of asynchronous circuits. A framework for automatic performance optimization of asynchronous systems is introduced, and an abstract performance model of the circuit on which the basic pipeline optimization problem can be defined is proposed. This model can be extended to support choices and is not restricted to deterministic pipelines [13]. An efficient Simulated Annealing algorithm is presented that demonstrates the feasibility of the optimization method for moderately sized models. The heuristic method is efficient, and produces solutions that achieve large performance speedup on large set of benchmarks. Systems are often designed with a target cycle time, while in this paper we study the problem of adding buffers to an asynchronous system to reduce the overall cycle time of the system with area and power overhead considerations. We introduce an abstract template model for handling slack matching problem that is comprehensible and capable to handle hierarchical circuit models and non-deterministic systems with a little extension. In addition, our technique is scalable for any size of problem and gives an acceptable solution in a reasonable time. Therefore, our technique is applicable on real systems. The remainder of this paper is organized as follows. Section 2 describes Timed Petri-Nets as the dominant performance analysis model. Section 3 discusses the Performance Optimization algorithm in detail while Section 4 gets on with the results and analysis. And finally Section 5 concludes the paper.

2 Performance Model

Our circuit model is based on Petri-Nets which have been already used as a description of concurrent systems [2]. The main advantage of our model is that it can be used for performance analysis in addition to simulation. For a general introduction to Petri Nets we refer the reader to [8].

2.1 Simple Templates

Different types of models can be developed for different kind of asynchronous templates as the target implementation style. We have developed a class of models that fully supports full buffer templates [6]. In the proposed model, Timed PetriNet (TPN), the detailed

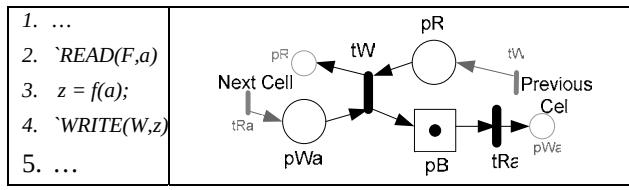


Fig. 1. A simple empty buffer modeled using Petri-Nets

structure of the original circuit including the handshaking channels are preserved which is the key that makes our model suitable for applications in which precise performance modeling is required. One important application is the slack matching [6] problem which our model can be readily applied to for addressing performance metrics.

The simplest form of an empty full buffer is a simple buffer that only reads a value from its input and writes it to its output. This behavior can be modeled simply as shown in Fig. 1. Transition tW is analogous to the write statement while place pWa emulates the write acknowledge. Similarly pR can be seen as the dual for read statement while tRa is the corresponding acknowledge. We considered delays on the place therefore forward delay and backward delay can be put on pR and pB. Full buffer is the same. But the token is at the pR place instead of pB place in the empty buffer. This model is very similar to FCBN model presented in[7] and the only difference is that we added tRa. The reason for this is that the used definition of the hierarchical Petri-Nets has a restriction on the input and output ports; all outputs must be transitions and all inputs must be places. This convention ensures that unwanted choices or merged constructs can not be formed when connecting Petri-Net modules to each other. The model for simple buffer can be extended to more reads and more writes.

It is notable that this form of modeling needs the least amount of transitions and places, as it requires only one place and one transition for each Read or Write basically. Additional constructs are needed when special functions like forking or decision making are required. For choice support in this model, we must add choice places for conditional reads and writes and two transitions for a choice place that identifies the state of condition (taken or not)[13]. Our methodology for buffer assignment is not restricted to deterministic models and can be applied on circuits containing choices. This extension has not been used in this work and is under investigation.

3 Automatic Slack Matching (ASM)

In contrast, in this section, we describe our slack matching algorithm for performance optimization of asynchronous circuits. Our algorithm is based on the simulated annealing (SA) method[11]. Given a feasible PetriNet structure, we perturb (Insert Buffer) it to obtain another feasible PetriNet structure through a set of pre-defined SA operations. After perturbation, we perform the suitable cost function to obtain a structure with respect to the precedence constraints. Finally a performance analyzing with simulation of the circuit is invoked to evaluate the solution quality.

3.1 Initialization

We use a PetriNet Simulator[13] to simulate our PetriNet model of circuit to obtain the stable state of tokens. Furthermore, this simulation provides such information as throughput which is needed at next steps. As mentioned in[13], the time complexity of the used PetriNet Simulator is $O(P^3)$, where P is the set places in the PetriNet model.

3.2 Perturbation

For the slack matching of QDI circuits, we introduce a type of SA operations. At iteration step we insert some buffers in the circuit and delete some buffers from the circuit randomly.

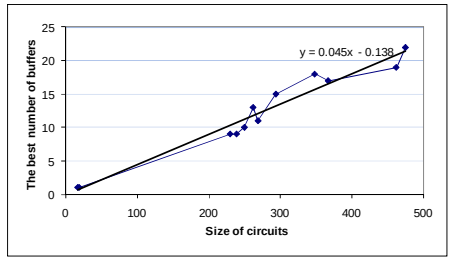


Fig. 2. The best number of buffers added per iteration

A) Buffer Insertion

The Run time of SA depends on the number of iterations of the algorithm, and will be prolonged if we add buffers into our circuit one by one. In order to decrease the run time, we add some pipeline buffers into the circuit simultaneously, per iteration. But the number of the added buffers per iteration must be calculated initially. Figure 2 shows the diagram that has been drawn with respect to the best number of the added buffers per iteration (y axis), and the size of the test cases(x axis). The short run time determines that how many buffers must be added per iteration. The equation of the best number of the added buffers per iteration with respect to the size of the circuit is y=0.0454x- 0.1386. So, we use this equation to determine the number of the buffers that must be added to the circuit per iteration.

B) Buffer Removing

The Perturbation step must have a deletion step. By this way, we let the algorithm remove any buffer that had been inserted in the circuit. At this step we delete one buffer from the circuit. This buffer can be any buffer that we added to the circuit from the start of the SA algorithm.

3.3 Cost Function

Our goal is to optimize Performance of the circuit with minimum penalties of area and power. So, parameters of the cost function are these terms:

- Performance: Defined throughput in asynchronous circuits. As mentioned, throughput is the inverse of the cycle time. The cycle time of a deterministic pipeline is defined as the largest cycle metric in its marked graph representation [1]. The cycle metric of each cycle is the sum of the delays of all associated transitions (or places) divided by the number of the tokens that can reside in the cycle. We use Karp method[9] to find the largest cycle metric of the circuits. This method creates a vertex for each marked place and edges between two vertices if there is a path between their corresponding places. The weight of an edge is the largest sum delay of such paths. Karp's algorithm will then find the maximum mean cycle.
- Area: A good approximation determined by the number of the cells in the circuit and their channels (transitions and places in the PetriNet structure). So, Sum of the nodes and their connections (places and transitions) is a good metric for area estimation.

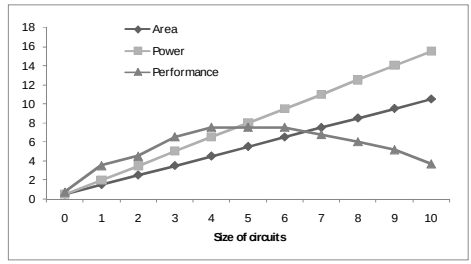


Fig. 4. Area, power and performance of a pipeline

- Power: Power in our work is estimated by transition counting presented in[14] that is applied on circuits after the decomposition step in the synthesis flow
- Figure 4 shows the area, power and performance increasing with respect to the size of an asynchronous pipeline with a constant token number. As shown in the figure, area and power increase linearly. But the performance increases initially by adding some buffers, and decreases afterwards by adding more buffers. Adding more buffers must stop when reaching the maximum point.

Therefore, the cost function Φ that optimized in our algorithm is given by

$$\Phi = \frac{\alpha \mathrm{A}/\mathrm{A}_i + \beta \mathrm{P}/\mathrm{P}_i + \gamma(\sum_j C_j / \sum_i C_i)}{\eta \mathrm{T}} \tag{1}$$

Where T is the throughput of the circuit, A is the total area of cells and communications, P is the estimated power using transition counting[14] and C is the delay of any cycle. So, ΣC is sum of cycles in the circuit. A_i, P_i and C_i are initial area, power and cycle length respectively. α, β, γ and η are user-specified constants and are used for normalization. Since we must improve performance, we penalize the excessive area and power. When SA minimizes the cost function, it automatically minimizes the penalty term. Thus, we can automatically optimize the circuits.Adding the buffers to the circuit may improve throughput or not. As known, the penalty terms are area and power. If the perturbation is not accepted, we undo the PetriNet structure changes.

4 Experimental Result

Fig. 5 shows the general structure of the proposed performance optimization scheme and its interface with a generic asynchronous synthesis flow. As shown, the design usually starts with a high level circuit specification. Verilog-CSP, standard Verilog powered by PLI[4], is considered here as the specification language. `READ and `WRITE blocking macros encapsulate asynchronous data communications in Verilog-CSP. By applying a set of functions preserving transformations to the high level specification, Data Driven Decomposition [5] method divides the original circuit to a set of communicating fine grain pipeline templates. Decomposition can potentially improve the overall performance by introducing more concurrency.

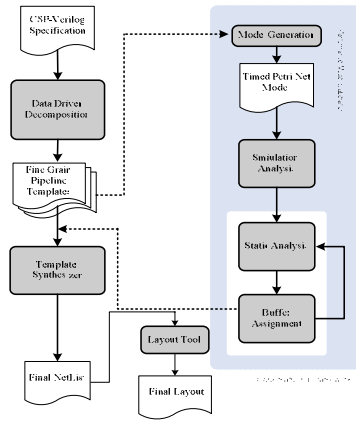


Fig. 5. Performance Evaluation Framework and its interface with asynchronous synthesizer

For optimization purposes, the output of the decomposition block which is in CSP format serves as input to a model generator that transforms a Verilog-CSP specification to its PetriNet equivalent. Then, our simulator runs the PetriNet circuit and provides the dynamic information of the original circuit such as throughput and token assignment. Our optimizer includes a static analyzer in order to provide static information that is needed in cost function evaluation, and a buffer assignment engine that adds buffers to the circuit utilizing the mentioned heuristic method. Then, the simulator runs the resulted circuit again. The resulted throughput of the circuit is compared to its original throughput, which verifies that the optimization operation was successful. The resulted circuit in PetriNet format is converged to a slack matched version of the input circuit with Verilog-CSP specification and serves as input to the next step of the synthesis. We used the well-known sequential benchmark of synchronous circuits referred to as ISCAS'89, to better explore the usefulness of the proposed algorithms.

In the beginning, our synthesis tool automatically translates the circuit (in verilog) to its PetriNet equivalent. At this step, the number of nodes of a synchronous circuits increase, because all primary inputs and outputs are considered nodes in our model. Also, we can not have multi-fanout wires in asynchronous systems, so they are transformed to reconvergent fanouts. Next, our algorithm slack matches the translated circuit. Persia[3] is a QDI synthesis toolset that is employed to synthesize our benchmarks.

For simplicity, we set the delay of each original or added buffer to 1 and the backward latency to 4, giving a local cycle time of 5. The results are shown in Table 1. Column 2 identifies the number of gates in the synchronous circuit. Column 3 shows the number of nodes in the PetriNet model of asynchronous circuit. Column 4 depicts the circuit throughput before adding slack buffers. Column 5 identifies the number of buffers that algorithm insert to the circuit. Column 6 shows the throughput of the resulted circuit after buffer insertion. Column 7 shows the improvement of throughputs mentioned. Column 8 identifies area overhead resulted from this algorithm. As demonstrated by experimental results, our proposed technique can achieve on average 38% optimization for performance with 26% increased area as its cost. Run time of the algorithm reported in column 9.

Table 1. Performance optimization for asynchronous circuits

circuit	# gates	# nodes	throughput	# buffers	Resulted throughput	% throughput	% area over-head	Run time (s)	%area im-provement compare to an ILP-based method
c17	6	17	0.366	5	0.448	22.6	29.4	5	-21.7
c432	160	250	0.281	82	0.380	35.3	32.8	35	203.9
c499	202	262	0.203	69	0.337	66.1	26.7	41	228
c880	383	509	0.185	124	0.279	51.1	24.3	72	53.3
c1355	546	713	0.192	308	0.266	38.6	43.2	121	247.6
s27	13	19	0.314	8	0.458	45.9	42.1	7	2.3
s382	181	230	0.297	48	0.402	35.6	20.8	59	22.2
s400	186	239	0. 240	63	0.312	30.4	26.3	64	-22.2
s420	234	294	0. 196	92	0.268	37.1	31.3	93	62.4
s444	202	269	0.211	76	0.298	41.7	28.2	81	6.9
s641	398	462	0.272	49	0.342	25.8	10.6	128	-10.2
s713	412	475	0.208	71	0.277	33.5	14.9	136	-14.5
s820	294	367	0.348	64	0.470	35.3	17.4	102	-17.1
s832	292	348	0.339	57	0.441	30.1	16.3	98	-16
s1196	547	719	0.175	246	0.257	47.2	34.2	174	256.6

A comparison is presented between our method and another method that was based on ILP formulation to minimize the cost of additional buffers in order to achieve a performance target [7]. The results are shown in column 10. Our method represents 65% improvement in reducing the area overhead; also it is much simpler than ILP formulation. . As shown in the table 1, our algorithm is better applied on the large circuits that need too many buffers for slack matching optimization. *C1355* test case is an example of this problem. ILP formulation insert 1800 buffers to the circuit but our algorithm obtained 38% performance improvement by the insertion of 308 buffers to the circuit. On the other hand, ILP formulation is better applied on the circuits that need a few buffers for slack matching. The ILP formulation solves the optimization problem for the *S820* circuit by addition of only one buffer to the circuit but our algorithm inserts 64 buffers to the test case.

5 Conclusion

In this paper, an efficient method presented for slack matching of asynchronous systems. The Decomposed circuit in the CSP-Verilog format is used to generate a Timed Petri-Net model which captures the dynamic behavior of the system. To obtain a high precision performance optimization, our scheme first performs a dynamic performance analysis to compute the situation of tokens. The proposed method is based on Simulated Annealing algorithm. The experimental results on a set of ISCAS benchmarks show that our proposed technique can achieve on average 38% optimization for performance, while there is 26% Are penalty. Considering the fact that better buffer

assignment can lead to better optimization with less area penalty, the future work is to improve the static analyzer. Also this work can be extended to support choice in the system.

References

1. Burns, S.M., Martin, A.J.: Performance Analysis and Optimization of Asynchronous circuits. In: Advanced Research in VLSI conference, Santa Cruz, CA (1991)
2. Kim, S.: Pipeline Optimization for Asynchronous circuits. PHD Thesis, University of Southern California (2003)
3. http://www.async.ir/persia/persia.php
4. Seifhashemi, A., Pedram, H.: Verilog HDL, Powered by PLI: a Suitable Framework for Describing and Modeling Asynchronous Circuits at All Levels of Abstraction. In: Proc. Of 40th, DAC, Anneheim, CA, USA (2003)
5. Wong, C.G.: High-Level Synthesis and Rapid prototyping of Asynchronous VLSI Systems. PhD's thesis, Caltech institute of technology (2004)
6. Lines, A.M.: Pipelined asynchronous circuits. Master's thesis, California Institute of Technology, Computer Science Department, CS-TR-95-21 (1995)
7. Beerel, P.A., Kim, N.H., Lines, A., Davies, M.: Slack Matching Asynchronous Designs. In: Proceedings of the 12th IEEE International Symposium on Asynchronous Circuits and Systems, Washington, DC, USA (2006)
8. Peterson, J.L.: Petri Net Theory and the Modeling of Systems. Prentice-Hall, Englewood Cliffs (1981)
9. Karp, R.M.: A Characterization of the Minimum Cycle Mean in a Diagraph. Discrete Mathematics 23, 309–311 (1978)
10. Prakash, P., Martin, A.J.: Slack Matching Quasi Delay-Insensitive Circuits. In: Proc. Of the 12th IEEE International Symposium on Asynchronous Circuits and Systems. IEEE Computer Society press, Los Alamitos (2006)
11. Kirkpatrik, S., Gellatt, C.D., Vecchi, M.P.: Optimization by Simulated Annealing. Science 220(4598) (1983)
12. Venkataramani, G., Goldstein, S.C.: Leveraging Protocol Knowledge in Slack Matching. In: Proc. of ICCAD 2006, San Jose, CA (2006)
13. Najibi, M., Niknahad, M., Pedram, H.: Performance Evaluation of Asynchronous Circuits Using Abstract Probabilistic Timed Petri Nets with Choice. ISVLSI (2007)
14. Ghavami, B., et al.: A Fast and Accurate Power Estimation Methodology for QDI Asynchronous Circuits. PATMOS, 463–473 (2007)

A Low Power SRAM Based on Five Transistors Cell

Arash Azizi Mazreah[1] and Mohammad Taghi Manzuri Shalmani[2]

[1] Islamic Azad University, Sirjan Branch
[2] Assistant Professor of Sharif University of Technology
aazizi@iausirjan.ac.ir, manzuri@sharif.edu

Abstract. This paper proposes a low power SRAM based on five transistor SRAM cell. Proposed SRAM uses novel word-line decoding such that, during a read/write operation, only selected cell is connected to bit-line when one row is selected whereas, in conventional SRAM (CV-SRAM), all cells in selected row connected to their bit-lines, which in turn develops differential voltages across all bit-lines, and this makes energy consumption on unselected bit-lines. Proposed SRAM uses one bit-line and thus has lower bit-line leakage compared to CV-SRAM. Furthermore, the proposed SRAM incurs no area overhead, and has comparable read/write performance versus the CV-SRAM. Simulation results in standard 0.25µm CMOS technology shows in worst case proposed SRAM has on average 80% smaller energy consumption in each cycle compared to CV-SRAM. Besides, energy consumption in each cycle of proposed SRAM and CV-SRAM investigated analytically, the results of which are in good agreement with the simulation results.

Keywords: SRAM, Write Operation, Read Operation, Capacitances, Dynamic Energy Consumption.

1 Introduction

Due to the high demands on the portable products, energy consumption is a major concern in VLSI chip designs. Specially, the low power static random access memory (SRAM) becomes more important because the number of memory cells and the bit width continue to increase.

A six Transistor SRAM cell (6T SRAM cell) is conventionally used as the memory cell. However, conventional SRAMs (CV-SRAM) based on this cell has large word-line and data-line capacitances, specially when the memory array size is high, thus in high frequency operation dynamic power consumption is very large and this is not useful for long time battery operation in mobile appliances. Since, in each column of memory array in CV-SRAM there are two bit-lines, therefore there are two leakages current from pre-charged bit-lines to ground through the access transistors of cells [1].

One major source of dynamic energy consumption in CV-SRAM is, when word-line asserted in one row of memory array, all cells in row connected to their bit-lines, which in turn develops differential voltages across all bit-lines, whereas one cell of connected cells to their bit-lines selected by column decoder for a read/write operation and other

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 679–688, 2008.

cells are unselected [2]. Conventionally hierarchical word-line decoding architecture is used to reduce the number of unselected cells to save power on unselected bit-line in CV-SRAM [3]. In this architecture, cells in each row divided into several groups and cells in selected group connected to their bit-lines only. But this architecture needs extra logics in each row of memory array and this, incurs considerable area overhead, especially when memory array size is high. Furthermore, in hierarchical word-line decoding architecture all cells in selected group connected to their bit-lines and this causes energy consumption in unselected bit-line in each group.

In this paper we describe new architecture for SRAMs based on five transistors SRAM cell (5T SRAM cell). This new architecture uses novel word-line decoding such that when word-line asserted in one row of memory array, only selected cell connected to bit-line. Hence dynamic energy consumption decreases. New architecture is based on a five transistor SRAM cell (5T SRAM cell). 5T SRAM cell used one bit-line and one access transistor, thus new architecture has low bit-line leakage [4].

2 Low Power SRAM Architecture

Fig. 1 shows the architecture of one column in proposed SRAM and Fig. 2 shows overall architecture of proposed SRAM. New SRAM uses novel word-line decoding. In this novel word-line decoding for each cell in memory array, we add one PMOS transistor with high threshold voltage. Our objective of additional transistor is that when word-line asserted in a row of memory array for selecting one cell during read/write operation, only selected cell connected to its bit-line as shown in Fig. 1. Therefore one cell connected to its bit-line if additional PMOS transistor turned on and corresponding word-line asserted to high voltage. This leads very low energy consumption during a read/write operation. Also the threshold voltage of additional PMOS transistor must be chosen as high as possible for smaller bit-line leakage. Since, new SRAM uses 5T cell, the additional transistors incur no area overhead compared to CV-SRAM with the same size.

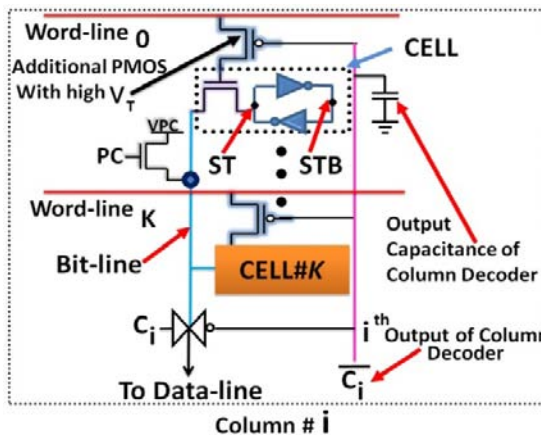


Fig. 1. Architecture of one column in proposed SRAM. (PC is pre-charge control).

The 5T cell has only one access transistor and a single bit-line. The writability of the cell is ensured by different cell sizing strategy [4]. In this cell for nondestructive read operation the bit-line is pre-charged to 1V [4]. Intermediate pre-charge voltage, 1V requires an on-chip DC-DC converter or external power supply [4].

2.1 Write Operation

When a write operation is issued the memory array will go through the following steps:

1-Row and column address decoding: the row address decoded for selecting a word-line. Also column address decoded for connecting a selected bit-line to data-line. 2-Bit-line driving: For a write, this bit-line driving conducts simultaneously with the row and column address decoding by turning on proper write buffer. After this step, selected data-line and bit-line will be forced into '1' or '0' logic level. 3-Cell flipping: If the value of the stored bit in the target cell is opposite to the value being written, then cell flipping process will take place. 4-Pre-charging: At the end of the write operation all bit-lines and data-line are pre-charged to 1V and memory array gets ready for next read/write operation.

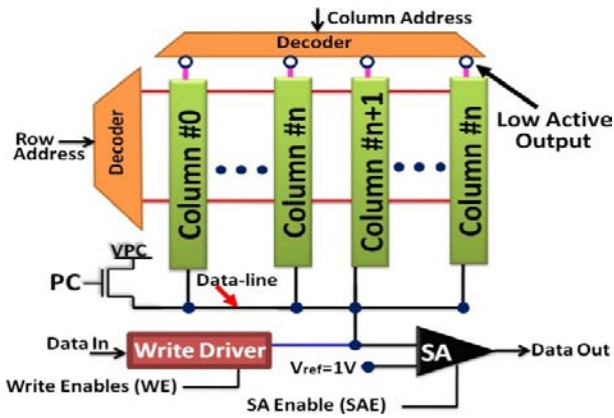


Fig. 2. Architecture of proposed SRAM. (PC is pre-charge control, SA is sense amplifier).

2.2 Read Operation

When a read operation is issued the memory will go through the following steps:

1-Row and column address decoding: the row address decoded for selecting a word-line. Also column address decoded for connecting a selected bit-line to data-line. 2-Bit-line deriving: After the word-line went to high voltage, the target cell connects to its bit-line. The so-called cell current through the driver or load transistor of target cell will discharge or charged the voltage of bit-line progressively, and this resulted a change on bit-line voltage and consequently on selected data-line. 3-Sensing: After word-line returned to low voltage, the sense amplifier (SA) is turned on to amplify the

small difference voltage between data-line and Vref=1V into full-swing logic signal. 3-Pre-charging: At the end of read operation all bit-lines and data-lines are pre-charged to 1V and memory array gets ready for next read/write operation.

3 Layout Design

Fig. 3 shows possible layout of 5T SRAM cell with proposed additional PMOS transistor in MOSIS scalable CMOS design rules. Also for comparison, Fig. 4 shows layout of 6T SRAM cell and 5T SRAM cell with additional PMOS transistor in MOSIS scalable CMOS design rules. The 6T cell has the conventional layout topology and is as compact as possible. 6T SRAM cell requires 25.3µm2 area in 0.25µm technology, whereas 5T SRAM cell with additional PMOS transistor requires 25µm2 area in 0.25µm technology. Also we design custom layout for two memory arrays with size 1Kbit for proposed SRAM and CV-SRAM in MOSIS scalable CMOS design rules as shown in Fig. 5. The memory array of proposed SRAM requires 27.02mm2 area in 0.25µm technology, whereas memory array of CV-SRAM requires 27.34mm2 area in 0.25µm technology. Therefore proposed SRAM incurs no area overhead compared to CV-SRAM with same size. Also all layouts designed by using Tanner EDA L-Edit layout tool.

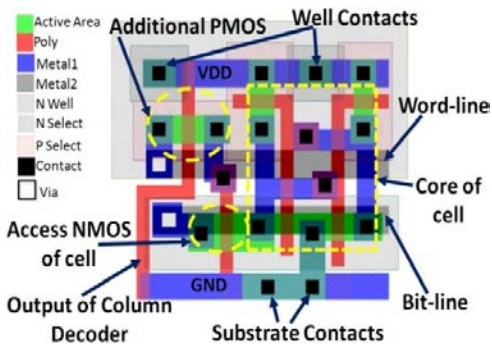


Fig. 3. Possible layout of 5T SRAM cell with additional PMOS

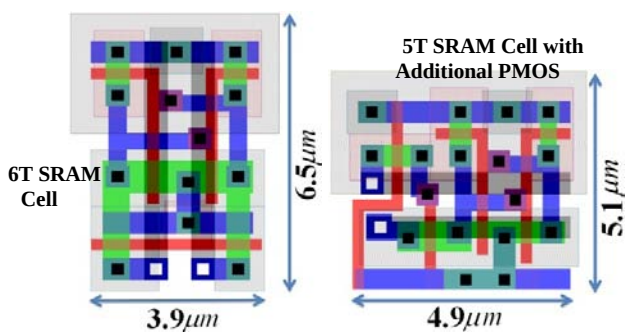


Fig. 4. Layout of 6T SRAM cell and 5T SRAM cell with additional PMOS

4 Capacitance in SRAMs

There are four premier parasitic capacitances in CV-SRAM and new proposed SRAM. These capacitances include bit-line ($C_{BL\text{-}New}$ and $C_{BL\text{-}CV}$), word-line ($C_{WL\text{-}New}$ and $C_{WL\text{-}CV}$), data-line ($C_{DL\text{-}New}$ and $C_{DL\text{-}CV}$) and output of column decoder ($C_{Out\text{-}Decoder}$). Later capacitance in new SRAM is considerable and we consider this capacitance only for new SRAM. Also we ignored wiring capacitance of bit-line, data-line and word-line in all expressions. The bit-line capacitance in CV-SRAM and new SRAM is mainly composed of the drain junction capacitance of pass transistors and this capacitance estimates by equation (1):

$$C_{BL\text{-}CV}=C_{BL\text{-}New}=C_J\times 2^{Row} \tag{1}$$

Where, C_J is drain junction capacitance of pass transistor of memory cells and *Row* is number of address lines in row decoder. The next large capacitance in CV-SRAM and new SRAM is word-line capacitance. In CV-SRAM this capacitance is mainly composed of gate capacitance of the pass transistor of memory cell and in new SRAM this capacitance is mainly composed of junction capacitance of additional PMOS transistors. This capacitance is estimated by expressions (2) and (3).

$$C_{WL\text{-}CV}=2\times C_g\times 2^{Column} \tag{2}$$

$$C_{WL\text{-}New}=C_{J\text{-}Add\text{-}P}\times 2^{Column} \tag{3}$$

Where, C_g is gate capacitance of pass transistor of memory cells, $C_{J\text{-}Add\text{-}P}$ is junction capacitance of additional PMOS transistor, and *Column* is number of address lines in column decoder. The next large capacitance in CV-SRAM and new SRAM is data line capacitance and this capacitance is mainly composed of junction capacitance of access column transistors and is estimated by expressions (4) and (5):

$$C_{DL\text{-}CV}=2\times C_{J\text{-}ColumnAccess}\times 2^{Column} \tag{4}$$

$$C_{DL\text{-}New}=2\times C_{J\text{-}ColumnAccess}\times 2^{Column} \tag{5}$$

Where, $C_{J\text{-}ColumnAccess}$ is drain junction capacitance of column access transistors and *Column* is number of address lines in column decoder. Finally the next capacitance that is considerable in new SRAM is the output capacitance of column decoder. This capacitance is due to gate capacitance of additional PMOS transistors (Fig. 1) and is estimated by expression (6):

$$C_{Out\text{-}Decoder}=C_{g\text{-}Add\text{-}P}\times 2^{Row} \tag{6}$$

Where, $C_{g\text{-}Add\text{-}P}$ is gate capacitance of additional PMOS transistor, and *Row* is number of address lines in row decoder.

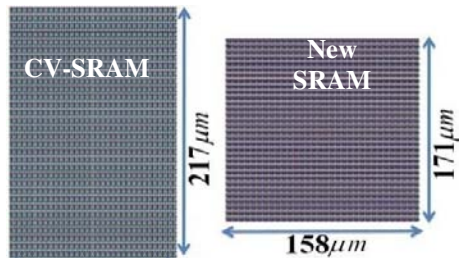


Fig. 5. Fully custom layout for two Memory arrays

5 Dynamic Energy Consumption in SRAMs

In SRAMs, either a read or write operation performs in each cycle. Therefore dynamic energy consumption in SRAMs consumes due to the charging and discharging capacitances during a read/write operation. Thus during each cycle of SRAMs a certain amount of energy is drawn from the power supply and dissipates. The amount of energy consumption in each cycle depends on type of operation in current cycle. Furthermore, when the capacitor CL charged from GND to VDD and then discharged to GND, amount of energy is drawn from the power supply and dissipated, equals to CLVDD2 [5][6] and stored energy on the capacitor CL with voltage VC equals to $\frac{1}{2}C_L V_C{}^2$. Thus each time the capacitor CL charged from VC to VDD and then discharged to VC amount of energy drawn from the power supply and dissipated, obtains by expression (7):

$$E_{Supply}=C_L(V_{DD}{}^2 - V_C{}^2) \tag{7}$$

In CV-SRAM, the best case of energy consumption of current cycle occurs when a read operation performs. Thus during current cycle (a read operation) of CV-SRAM word-line charged to VDD and then discharged to GND and bit-lines and data-line from VDD discharged to VBL-Read and then charged to VDD. Therefore the best case energy consumption after each cycle in CV-SRAM is estimated analytically by expression (8):

$$\begin{aligned} E_{BC\text{-}CV} \cong\ & C_{WL\text{-}CV}V_{DD}{}^2 + C_{DL\text{-}CV}\times(V_{DD}{}^2 - V_{BL\text{-}Read}{}^2) \\ & + 2^{Column}\left[C_{BL\text{-}CV}\times(V_{DD}{}^2 - V_{BL\text{-}Read}{}^2)\right] \end{aligned} \tag{8}$$

Where, $V_{BL\text{-}Read}$ is voltage of bit-lines and data-line just after activation of the word-line in a read operation. In new SRAM, the worst case of energy consumption of current cycle occurs when in current cycle a write operation with data '1' will be performed. Thus during current cycle (write operation with data '1') of new SRAM, word-line charges to V_{DD} and then discharges to GND and output of column decoder charged to V_{DD} and discharged to GND, and selected bit-line and data-line charges

from V_{PC} to V_{DD} and then discharges to V_{PC}. Thus the analytically worst case energy consumption after each cycle in new SRAM obtained by following expression:

$$E_{WC-New} \cong (C_{WL-New} + C_{out-Decoder}) \times V_{DD}^{2} + (C_{DL-New} + C_{BL-NEW}) \times (V_{DD}^{2} - V_{PC}^{2}) \quad (9)$$

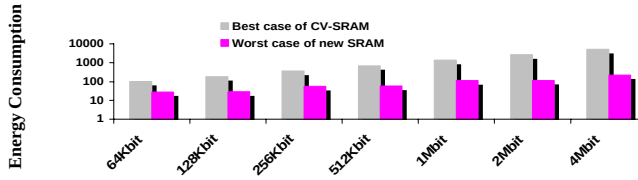


Fig. 6. Analytical best case and worst case energy consumption in each cycle of CV-SRAM and new SRAM

Fig. 6 shows the worst case energy consumption of new SRAM and best case energy consumption of CV-SRAM in each cycle according to size of memory array in standard 0.25µm CMOS technology and with parameters listed in Table 1. As shown in Fig. 6 energy consumption after each cycle of new SRAM is very smaller than CV-SARM. On the average analytically worst case energy consumption of new SRAM is %89 smaller than analytically best case energy consumption of CV-SRAM.

6 Simulation Results

To verify correct operation of new SRAM and delay comparison with CV-SRAM, we simulate a new SRAM and CV-SRAM with size 256Kbit using HSPICE in standard 0.25µm CMOS technology with 2.5V for supply voltage and 0.5V for threshold of additional PMOS transistor. For optimizing the delay of row and column decoders pre-decoding scheme has been used in proposed SRAM and CV-SRAM [1]. Based on layouts shown in Fig. 4, all parasitic capacitances and resistances of bit-lines, data-line and word-lines are included in the circuit simulation. Fig. 7 shows circuit schematic of the sense amplifier used in new SRAM and CV-SRAM and Fig. 8 shows circuit schematic of the write driver used in new SRAM and CV-SRAM uses convectional write driver. For testing the correctness of a read and write operation of new SRAM, following scenario applied to new SRAM:

- Writing ‘0’ in to one cell and then read it
- Writing ‘1’ in to one cell and then read it

Fig. 9 show simulated waveform with applying above scenario. Table 2 shows delay and SNM (static noise margin) comparisons of new SRAM and CV-SRAM.

Also to evaluate the effectiveness of new SRAM for saving dynamic energy consumption and precision of derived analytical expressions above, based on layouts shown in Fig. 4 we simulated a read and write operation of new SRAM and

Table 1. Parameter values used in analytical expressions

Symbol	Parameter	Value
V_{DD}(V)	Supply voltage	2.5
V_{PC}(V)	Pre-charge voltage	1
$V_{BL\text{-}Read}$(V)	Voltage of bit-lines after word-line activation during read in CV-SRAM	2.25

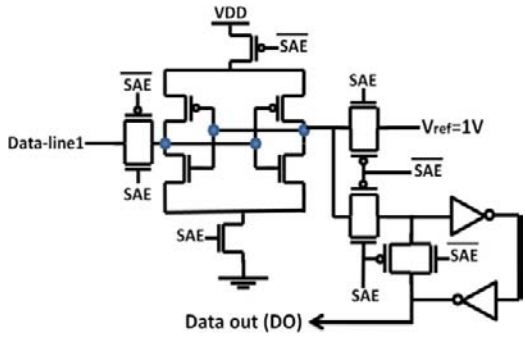


Fig. 7. Circuit schematic of sense amplifier [5]

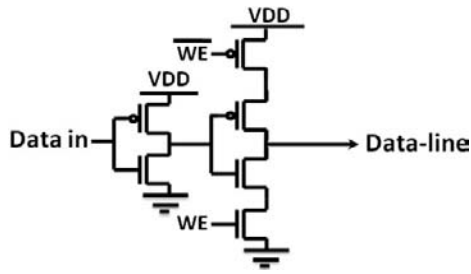


Fig. 8. Circuit schematic of write driver

CV-SRAM for different size of memory array in standard 0.25µm CMOS technology. To measure the best case energy consumption of CV-SRAM, a read operation is performed and then amount of energy is drawn from power supply and dissipated is measured. For worst case energy consumption in new SRAM, a write operation with data '1' is performed and then amount of energy is drawn from power supply and dissipated is measured. These measurements are shown in Fig. 10. Simulated worst case energy consumption of new SRAM is %80 smaller than simulated best case energy consumption of CV-SRAM on the average. Thus the simulation results are in good agreement with the analytical expressions and proposed new SRAM has great potential to save the dynamic energy consumption, especially when dimensions of memory array are large. Compared to the CV- SRAM, the SNM of the new SRAM is about 50% lower. The reduced SNM is a disadvantage. However, the SNM level is still sufficiently large to support correct functionality of the memory cell across the process corners.

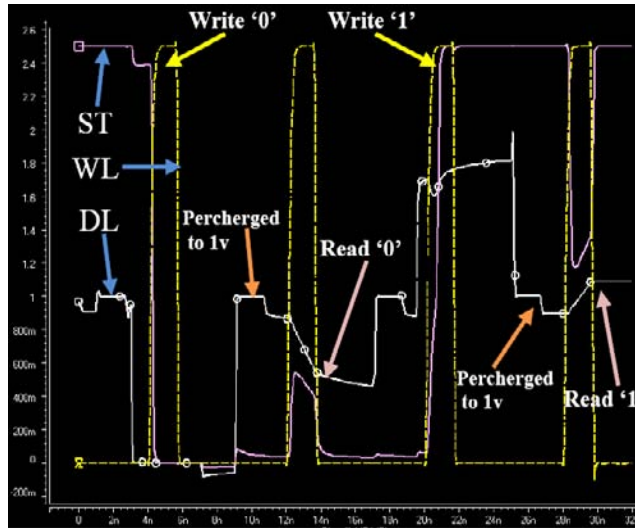


Fig. 9. Simulated waveform of read/write operation (WL is word-line, and DL is Data-Line and ST specified in Fig. 1)

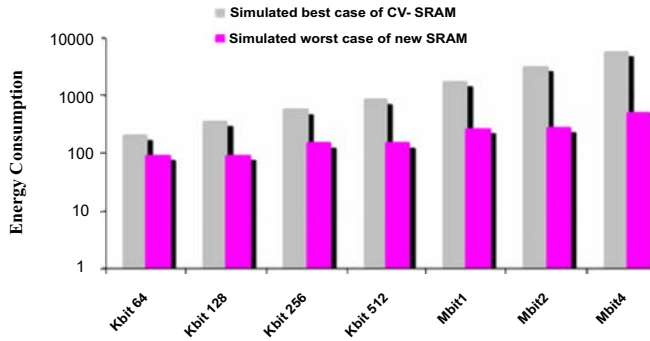


Fig. 10. Simulated best case and worst case energy consumption in each cycle of CV-SRAM and new SRAM

Table 2. Delay and SNM comparison of SRAMs

Metrics	CV-SRAM	New SRAM
Read delay (word-line high to equalization)	**2.1 ns**	**2.2ns**
Write delay (data-line driving to equalization)	**1.6ns**	**2ns**
SNM	**1.51V**	**0.76V**

7 Conclusion

This paper presents a low power SRAM. In newly SRAM for each column there is one bit-line. We add one PMOS transistor with high threshold voltage for each cell. The purpose of adding these PMOS transistors is that when one row selected in new SRAM, only on cell connected to bit-line and thus dynamic energy consumption is reduced. Since we use 5T cell there is not any overhead constrain compared to conventional SRAM with the same size. Furthermore in this paper we present explicit analytical expressions for estimating the capacitances and energy consumption in proposed SRAM and CV-SARM. Simulation results in standard 0.25µm CMOS technology show that this new SRAM have great potential to save the dynamic energy consumption. Finally the proposed SRAM has comparable read/write performance versus CV-SRAM.

References

1. Amelifard, B., Fallah, F., Pedram, M.C.: Low-leakage SRAM design with dual Vt transistors. In: 7th IEEE International Symposium on Quality of Electronic Designs. IEEE Press, Los Alamitos (2006)
2. Martin, K., C.: Digital Integrated Circuit Design. Oxford university press, New York (2000)
3. Hirose, T., Kuriyama, H., Mnmkami, S., Yuzuriha, K., Mukai, T., et al.: A 20-ns 4-Mb CMOS SRAM with Hierarchical Word Decoding Architecture. IEEE Journal of Solid-State Circuits 25, 1068–1074 (1990)
4. Carlson, I., Andersson, S., Natarajan, S., Alvandpour, A.C.: A high density, low leakage, 5T SRAM for embedded caches. In: Proc. of European Solid-State Circuits Conference, pp. 215–222. IEEE Press, Los Alamitos (2004)
5. Rabaey, J.M., et al.: Digital Integrated Circuits: A Design Perspective. Prentice Hall, Englewood Cliffs (2002)
6. Azizi Mazreah, A., Manzuri Shalmani, M.T., Barati, H., Barati, A.C.: Delay and Energy Consumption Analysis of Conventional SRAM. International Journal of Electronics, Circuits and System 2(2), 35–39 (2008)

Evaluating the Metro-on-Chip Methodology to Improve the Congestion and Routability

Ali Jahanian, Morteza Saheb Zamani, Mostafa Rezvani, and Mehrdad Najibi

Information Technology and Computer Engineering Department
Amirkabir University of Technology,Tehran, Iran
{jahanian,szamani,m.rezvani,mnajibi}@aut.ac.ir

Abstract. Recently, metro-on-chip technique has been presented to improve the congestion of design by serializing parallel wires via delay insensitive asynchronous serial transceivers. In this paper, metro-on-chip technique is improved in terms of routability and computation time and also its impact on routability is examined. Finally, it is evaluated in 130nm technology. Experimental results show that for attempted benchmarks, congestion and routability are improved by 15.54% and 21.57% on average, respectively. Total wire length is reduced up to 6.3% with a slightly increasing in power consumption (0.12% on average).

Keywords: Asynchronous serial transmission, congestion, routability.

1 Introduction

Asynchronous serial transceivers have been used to improve the congestion and routability in application specific integrated circuit design methodology. With the emerging of the globally asynchronous locally synchronous concept [1], some techniques for using asynchronous serial transceivers in network-on-chip (NOC) systems have been proposed [2]. These systems were mainly used to improve the data transmission delay between cores. Conventional asynchronous circuits used differential-signaling serial links that made them inappropriate for on-chip applications. This is because they require PLL-based clock and data recovery circuits which consume excessive power and area [3].

In [4], an NOC with a new serialization mechanism was proposed which provides high performance serialization with small overhead. In this approach, the data transmission bandwidth is automatically adjusted based on the workload of the network. This method was implemented and tested in 0.18μm technology at 3Gb/s. In [4], the authors proposed a serial link transceiver for global on chip communications working at 3Gb/s per wire in a standard 0.18μm CMOS process with low crosstalk-induced delay variability. They implemented pure electrical link without any repeater in higher layers of metals. Teifle and Manohar [5] presented a high-speed and clock-less serial link transceiver for inter-chip communication. Their experiments showed that their transceiver operates at up to 3 Gb/s in 0.18μm CMOS technology. Dobkin et al. [6] presented a novel bit-serial interconnect structure with a new data encoding style

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 689–696, 2008.

to encode each bit in one transition in a pipelined structure. They showed that their asynchronous circuit can send or receive a new bit at a data cycle of a single FO4 (i.e. fan-out of 4) inverter delay.

Authors of this paper presented a new approach for serializing the parallel wires using asynchronous serial transceiver to improve the routing congestion of design [7]. This mechanism was called metro-on-chip (MOC) because its semantic is similar to metro transportation systems in large cities which convey many passengers in one trip. It is worthwhile to note that congestion metric is a good metric while can be more useful if it is measured in terms of the routability. However in [7] there is no report about the routability of design after MOC insertion.

This paper is organized as follows. Section 2 describes the concept of the metro-on-chip briefly. In Section 3, the proposed MOC-based design flow is presented. Experimental results are reported in Section 4 and finally, Section 5 concludes the paper.

2 Overview on the Metro-on-Chip

In the conventional physical design flow, terminals of each net are connected via dedicated and specialized wires. However, in new circuits which have large number of nets, routing congestion raised issues in terms of routability, signal integrity and even performance of the design. Multiplexing the nets in congested circuits can be an effective technique to congestion reduction and routability improvement. The main concept of this paper is to automatically find non-critical and sufficiently long wires and multiplex and send them using asynchronous serial transmission hardware. This idea is similar to the metro system in large cities that conveys many passengers in one trip to reduce traffic congestion. In this paper, a static paradigm of metro is implemented in which the metro has fixed stations at the source and the destination without any intermediate stations [7].

Fast serial communication mechanism proposed in [6] is used as the serial link hardware. The area, delay and other characteristics of this system have a key role in the feasibility and quality of the proposed EDA design flow. The fast serial link transceiver [6] uses low-latency synchronizers at the sources and sinks with a two-phase non-return the zero level encoded dual rails (LEDR) asynchronous protocol that allows non-uniform delay intervals between successive bits. Acknowledgment of transmission is returned only once per word, rather than bit by bit. This enables multiple bits to be transmitted in a wave-pipelined manner over the serial channel.

In [6], a high data rate shift-register structure for asynchronous serializer and deserializer has been proposed. It operates at about one fan-out 4 inverter (FO4) delay and can be fully implemented in CMOS technology without any requirement to PLL or auxiliary clock generation circuitry.

Let MOC cluster i consists of m_i wires. The serial transmission hardware of this cluster contains m_i shift-registers in MUX module and also m_i shift-registers in DE-MUX module. With a good clustering algorithm, source/sink nodes are close to the source/sink gravity centers and the delays of input/output wires are very small. Because of the pipeline structure of the serial link transceiver proposed in [6], each input signal can be applied to the MUX input while the previous signal is being multiplexed (except for the first input). Moreover, each output signal can be applied to its sink

while the previous signal is being demultiplexed (except for the last output). Therefore, total delay of the i^{th} MOC cluster with n wires can be calculated as:

$$D_{(i)} = ID(i,1) + 2.m_i.\max(D_{FO4}, TD(i)) + OD(i, m_i) \quad (1)$$

where *ID(i,1)* is the delay of the *first* input signal, *OD(i,m_i)* is the delay of the *last* output, *TD(i)* is the transmission line delay in i^{th} cluster and D_{FO4} represents an FO4 inverter delay.

Inserting an asynchronous module in a synchronous circuit may change the functionality of the whole system due to the violation of the setup and hold time requirements of synchronous registers. In [7], a feasible solution for interfacing the asynchronous transmission module and a synchronous circuit with a global clock is presented. In proposed mechanism, the clock period must be extended by the delay of MOC hardware if the paths containing the multiplexed wires do not have enough delay slack (i.e. their delays are close to the maximum clock period of the design). However, if the paths of the selected wires have enough delay slack, the multiplexing delay will not affect the critical path of the design.

3 MOC-Based Physical Design Flow

In the proposed design flow, MOC modules are inserted after detailed placement where the position of cells have been determined and delay estimations are not too rough. The details of proposed MOC insertion design flow are described in the following sub-sections.

3.1 Floorplanning and Placement

In this stage, the design is floorplanned and the final locations of cells are determined. The quality of the placement algorithm has a great impact on the congestion and wirelength benefits of MOC system. We used the Dragon [8] for floor-placement of attempted benchmarks.

3.2 Static Timing Analysis

In this phase, a static timing analysis (STA) is performed to estimate path delay. After estimating the topology of nets, the delay of each path in the design is calculated using Elmore delay model and the critical paths of the design are determined. Critical path list will be used to select non-critical nets for multiplexing.

3.3 Select Non-critical Nets

In this stage, a list of non-critical and sufficiently long wires is generated based on the results of the static timing analysis. After STA, a list of paths whose delays are greater than a specified threshold (T_d) has been generated. These nets cannot be multiplexed because they may violate the timing constraints of the design. Therefore, these nets are marked to be excluded from MOC groups.

3.4 Cluster the Multiplexed Nets into MOC Groups

In this phase, the selected nets are clustered into some groups such that the nets in each group have close sources and close sinks. Let $G=\{g_1, g_2, \ldots, g_n\}$ be a set of MOC clusters where cluster g_i has m_i nets. *Companionship* of g_i, denoted as *CS(i)*, is defined as the cost of multiplexing the nets in g_i. *CS(i)* can be calculated as *SrcD(i)+SnkD(i)* where *SrcD(i)*/*SnkD(i)* is the summation of distances between sources/sinks in g_i. The developed clustering algorithm which uses Fibonacci min-heap is shown in Fig 1.

Clustering Algorithm

```
1:  Create initial CS Fibonacci Min-heap
2:  WHILE (there are any uncommitted nets)
3:     Get minimum CS from FHeap
4:     Merge minimum CS pair to make new cluster
5:     Update the FHeap
6:     IF (size of cluster > MAX_CLUSTER_CAP)
7:        Mark the cluster as final cluster
8:     END
9:  END
10: Finalize the clusters
```

Fig. 1. Clustering algorithm for MOC insertion

At the start of the algorithm, each net is considered as a cluster with one net. First, a CS Fibonacci min-heap (FHeap) is generated showing the companionship cost of each pair of clusters. Then, two clusters with minimum CS are fetched from FHeap and merged in each loop of the algorithm and then the list of costs of cluster pairs in CS FHeap is updated. If the size of the merged clusters reaches a predefined offset, the cluster is marked as final cluster. The algorithm is continued until there are no unmarked clusters. The complexity of this algorithm is $O(n^2\log(n))$ which is better than the complexity of clustering in [7] ($O(n^3)$).

3.5 Repair the MOC Groups

After clustering the nets, clusters are evaluated in order to gain more benefits from MOCs and some MOC clusters are repaired to avoid the clock period violation.

3.6 Insert MOC Devices into Netlist

In this phase, multiplexer and demultiplexer cells are added to the netlist and it is updated to connect the MOC devices to the source and the sink nodes of the clusters.

3.7 Incremental Placement of MOC Devices

After inserting the MOC instances into the netlist, their locations in the layout must be determined. An incremental placement algorithm is used to determine the final locations of the MOC devices in the netlist.

4 Experimental Results

We implemented our algorithm in C++ on an Intel 4300 workstation with 2 GB of memory. It was applied to twelve circuits randomly selected from the IWLS-2005 benchmarks [9] with various design sizes from 3227 to 92048 cells. Table 1 shows the characteristics of the benchmarks together with the results of MOC insertion.

Table 1. Benchmarks characteristics

Index	Benchmark	#Cells	#MOCs	#Members	TW (ps)
1	Spi	3227	0	0	2524.24
2	tv80	7161	0	0	4163.09
3	pci_bridge32	16816	9	23.4	18727.58
4	dma	19118	11	24.5	18491.70
5	b20_1	19131	14	10.21	6961.51
6	b22	28317	24	18.75	10796.53
7	b22_1	28128	45	24.3	10233.41
8	wb_conmax	29034	0	0	8816.92
9	b17	37117	28	26.55	27207.41
10	b17_1	37383	36	36.86	27291.20
11	ethernet	46771	36	30.05	23330.71
12	b18	92048	24	96.7	73038.43

In this table, #Cells show the number of cells in each benchmark, *#MOC*s represents the number of generated MOC clusters in each design. Column *#Members* shows the average number of wires which are multiplexed in MOC clusters and *TW* represents the critical delay of the benchmarks. As can be seen in Table 1, circuits 1 and 2 have no long critical paths for MOC insertion and benchmark 8 has medium critical paths but its nets are not long enough. Therefore, our algorithm did not insert any MOC modules into these circuits. Two different design flows were attempted to test the proposed idea. In the first design flow, called conventional net routing (CNR), terminals are connected via dedicated wires and in the second flow (MOC), nets were selected and serialized asynchronously.

4.1 Congestion and Routability

For congestion estimation, the method proposed in [10] was used. Table 2 shows the results of our experiments on the benchmarks in terms of design congestion in 130nm and 180nm. *Overflow* shows the percentage of bins that have congestion level beyond the routing capacity of bins (congestion overflow) and *Congestion Improvement* represents the congestion reduction of MOC vs. CNR.

Table 2 shows that for small circuits, the improvements in 130nm is not more than those in 180nm but for large circuits, the benefits of MOC insertion in 130nm is considerably more that those in 180nm. Therefore, benefits of MOC mechanism are small. It is worth noting that congestion reduction in benchmarks 3 and 11 are higher that others since their netlist have large busses which generate very beneficial MOC clusters.

Table 2. Congestion before and after MOC insertion

Index	Benchmark	130 nm			180 nm		
		Overflow		Congestion Improvement (%)	Overflow		Congestion Improvement (%)
		CNR (%)	MOC (%)		CNR (%)	MOC (%)	
1	spi	2.47	2.47	0	-	-	-
2	tv80	7.37	7.37	0	-	-	-
3	pci_bridge32	1.9	1.51	20.5	0.85	0.66	28
4	dma	15.45	13.96	9.6	17.94	16.33	9.8
5	b20_1	48.32	46.39	3.9	36.3	34.4	5.2
6	b22	46.14	45.02	2.4	34.2	32.6	4.6
7	b22_1	61.43	52.86	13.9	58.1	52.3	9.9
8	wb_conmax	34.47	34.47	0	-	-	-
9	b17	6.70	5.48	18.2	4.7	4.1	12.7
10	b17_1	6.94	5.74	17.2	59.96	49.3	17.7
11	ethernet	11.27	7.98	29.1	59.96	54	19.35
12	b18	4.54	3.73	16.1	1.02	0.6	14
Average				**15.54**			**13.47**

Table 3. Design routability of benchmarks

Index	Benchmark	#Nets	Misroute		Routability Improvement (%)
			CNR (%)	MOC (%)	
1	spi	4100	0	-	0
2	tv80	7220	1	-	0
3	pci_bridge32	17122	34	29	14.7
4	dma	19640	41	31	24.3
5	b20_1	13896	104	84	19.2
6	b22	270120	346	284	17.9
7	b22_1	270034	220	190	13.6
8	wb_conmax	29321	22	-	0
9	b17	39233	303	243	19.8
10	b17_1	39622	281	202	28.1
11	ethernet	52816	583	375	35.6
12	b18	96842	632	501	20.7
Average					**21.57**

To evaluate the routability improvement in MOC compared with CNR, we used Atlas tool [11] as standard-cell global router. This tool routes the design by iterative rip-up & reroute operations. At the end of the rip-up & reroute process, mis-routed nets are routed by a post-process algorithm, if there are any. In this experiment, routing layers are restricted to only 2 layers in global routing stage in order to emphasis the benefits of MOC insertion. Table 3 shows the routability improvement of MOC insertion process. In this table, *#Nets* shows the number of nets in the netlist before MOC insertion, *Misroute* represents the percentage of misrouted nets before post-process phase of global routing and column *Routability Improvement* shows the improvement of routability of MOC compared with CNR. As can be seen in this table, routability of design is increased by 21.57% on average.

4.2 Wirelength Reduction

In Table 4, wirelength reduction after MOC insertion is reported. *TWL* represents the value of total wirelength and *TWL Reduction* shows the improvement in total wirelength. As can be seen in this table, the decrease in total wirelength is small because the number of candidate nets for MOC is much smaller than the total number of nets in a design.

Table 4. Experimental results for wirelength

Index	Benchmark	TWL		TWL Reduction (%)
		CNR (μm)	MOC (μm)	
1	spi	5.221e8	5.221e8	0
2	tv80	1.350e9	5.221e8	0
3	pci_bridge32	4.000e9	3.967e9	1.01
4	dma	5.762e9	5.652e9	1.94
5	b20_1	4.895e9	4.826e9	1.43
6	b22	6.482e9	6.457e9	0.38
7	b22_1	7.577e9	7.126e9	6.3
8	wb_conmax	6.324e9	6.324e9	0
9	b17	1.243e10	1.240e10	0.2
10	b17_1	1.248e10	1.167e10	6.9
11	ethernet	1.683e10	1.612e10	4.3
12	b18	3.269e10	3.092e10	5.5
Average				**3.08**

4.3 Power Consumption

Power consumption is an important concern in current design methodologies and any new design flow may need to consider it. Our experiments show that power consumption of the benchmarks after MOC insertion is slightly more than the initial state (0.12% on average). Note that the major part of the increased power is resulted from static power consumption of MOC cells.

4.4 Clustering Runtime Improvement

Our experiments show that clustering runtime is improved about 24.3% compared with the clustering runtime of [7] on average. Clustering runtime improvement is more considerable for large circuits that have more MOC clusters. This can make the mechanism more practical for large and congested circuits.

5 Conclusion

In this paper, a new approach for congestion reduction and routability improvement inspired by the idea of the metro transportation systems in large cities was proposed. In the proposed method, a set of wires are marked as non-critical and sufficiently long ones and then these wires are multiplexed at the source, transmitted via an asynchronous serial link and demultiplexed at the destination. Experimental results show that

the overflow congestion is reduced by 15.54% on average and routability and total wirelength are increased by 21.57% and 3.08% respectively, on average. On the other hand, the power consumption of the attempted benchmarks is increased slightly (0.12% on average). MOC insertion design flow may be improved if the distribution of long wires and white spaces for MOC insertion are planned in earlier stages of physical design flow such as floorplanning.

References

1. Carloni, L.P., Sangiovani-Vincentelli, A.L.: On-chip Communication Design: Roadblocks and Avenues. In: CODES+ISSS, pp. 75–76 (2003)
2. Bjerregaard, T., Mahadeven, S.: A Survey of Research and Practices of Network-on-Chip. ACM Computing Surveys 38(1), 1–51 (2006)
3. Meincke, T., et al.: Globally Asynchronous Locally Synchronous Architecture for Large High-performance ASICs. In: International Symposium on Circuits and Systems, pp. 512–515 (1999)
4. Lee, S.J., Kim, K., Kim, H., Cho, N., Yoo, H.J.: Adaptive Network-on-Chip with Wave-front Train Serialization Scheme. In: VLSI Circuits, pp. 104–107 (2005)
5. Teifle, J., Manohar, R.: A High-speed Clockless Serial Link Transceiver. In: International Symposium on Asynchronous Design, pp. 151–161 (2003)
6. Dobkin, R., Kolodny, A., Morgenshtein, A.: Fast Asynchronous Bit-serial Interconnects for Network-on-Chip. In: International Symposium on Asynchronous Design, pp. 117–127 (2006)
7. Jahanian, A., Saheb Zamani, M.: Metro-on-Chip: an Efficient Physical Design Technique for Congestion Reduction. IEICE Electronics Express 4(16), 510–516 (2007)
8. Taghavi, T., Yang, T., Choi, B.-K.: Dragon2005: Large Scale Mixed Size Placement Tool. In: International Symposium on Asynchronous Design, pp. 245–247 (2005)
9. IWLS Benchmarks, `http://iwls.org/iwls2005/benchmarks.html`
10. Saeedi, M., Saheb Zamani, M., Jahanian, A.: Prediction and Reduction of Routing Congestion. In: International Symposium on Physical Design, pp. 72–77 (2006)
11. STD-Cell PathFinder Global router (2007), `http://ceit.aut.ac.ir/~eda`

Sequential Equivalence Checking Using a Hybrid Boolean-Word Level Decision Diagram

Bijan Alizadeh[1] and Masahiro Fujita[2]

[1] School of Electrical Engineering, Sharif University of Technology, Tehran, Iran
[2] VLSI Design and Education Center (VDEC), Universtiy of Tokyo, Tokyo, Japan
b_alizadeh@sharif.edu, fujita@ee.t.u-tokyo.ac.jp

Abstract. By increasing the complexity of system on a chip (SoC) formal equivalence checking has become more and more important and a major economical issue to detect design faults at early stages of the design cycle in order to reduce time-to-market as much as possible. However, lower level methods such as BDDs and SAT solvers suffer from memory and computational explosion problems to match sizes of industrial designs in formal equivalence verification. In this paper, we describe a hybrid bit- and word-level canonical representation called Linear Taylor Expansion Diagram (LTED) [1] which can be used to check the equivalence between two descriptions in different levels of abstractions. To prove the validity of our approach, it is run on some industrial circuits with application to communication systems and experimental results are compared to those of Taylor Expansion Diagram (TED) which is also a word level canonical representation [2].

Keywords: Sequential Equivalence Checking, Formal Verification, System on Chip (SoC) and Canonical Representation.

1 Introduction

Increasing in the size and complexity of digital systems (SoCs) has made it essential to address sequential equivalence verification issues at early stages of the design cycle. Sequential Equivalence Checking (SEC) is a process of formally proving functional equivalence of designs that may in general have sequentially different implementations. Examples of sequential differences span the space from retimed pipelines, differing latencies and throughputs, and even scheduling and resource allocation differences. In order to match sizes of real world designs in formal equivalence checking, reducing run times and the amount of memory needed for computations is a key point. Most of hardware verification tools however are based on bit-level methods like BDD or SAT solvers that suffer from size explosion problems when dealing with industrial designs.

On the other hand, the designs are usually given as *Algorithmic-Specification* in C (ASC) and *Register-Transfer-Level* (RTL) in HDL that comprise lots of arithmetic operations such as multiplications. If we employ BDD or Boolean SAT based methods to verify the equivalence of two descriptions, multiplications should be encoded into bit-level operations and therefore these techniques will fail because of too many numbers of Boolean variables or clauses. In addition arithmetic operations often have

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 697–704, 2008.

very regular structures that can be described easily on a higher level of abstraction. If verification procedures operate on the basis of bit-level descriptions, this information is lost on bit-level and thus cannot be utilized by verification tools. Hence the basic idea comes from using a word level representation that will have fewer variables and will be therefore better suited for equivalence verification. Furthermore we need a canonical representation of functions with a mixed Boolean and integer domain, and an integer range in order to verify equivalence of two high level descriptions. In the literature of canonical graph-based representation, we find a canonical word level representation called TED that is able to represent functions with an integer domain and range [2]. While other representations can only define integer-valued functions over binary variables as a bit vector, TED uses Taylor series as its decomposition method (for an overview see [2]). In contrast to BDDs and BMDs, TED is based on non-binary decomposition principle where the decomposition at each node leads to more than two terms. Although it is very suitable to represent polynomial expressions at algorithmic level descriptions, it requires more run time and memory to represent RTL designs that contain logical operations as well as arithmetic operations.

The reminder of this paper is organized as follows. Related works are discussed in Section 2. LTED as an integrated canonical representation is briefly described in Section 3. A methodology for use of LTED to check the equivalence between ASC and RTL descriptions is presented in Section 4. Experimental results are shown in Section 5.

2 Related Work

In this section we discuss earlier works that are related to our approach. Methods to verify the system level description against the RTL model are still evolving. Recently, some techniques have been proposed to apply equivalence checking to the system level and RTL descriptions [3-9].

A solution with a C-based bounded model checking (CBMC) engine was proposed in [7, 8] that takes a C program and a Verilog implementation. They described an innovative method to convert the C program, including pointers and nested loops, into Boolean formulas. The Verilog code is also converted to Boolean formulas by a synthesis-like process. Then the two programs are converted into a Boolean satisfiability problem. Since this tool works entirely in the Boolean domain the capacity of CBMC is limited by space and time considerations.

In [5] an equivalence checking technique to verify system level design descriptions against their implementations in RTL was proposed. It presented an automatic technique to compute high level sequential compare points to compare variables of interest in the candidate design descriptions. They start the two design state machines at the same initial state and step the machines through every cycle, until a sequential compare point is reached. At this point the equivalence of the two state machines is proved using a lower (Boolean) level engine which is zChaff SAT solver. One of limitations of this technique is not to be scalable in the number of cycles. As the number of cycles gets larger, the size of the expression grows quadratically, causing capacity problems for the lower level SAT engine. Furthermore it may not be applicable to large designs due to arithmetic encoding.

The authors in [6] have proposed early cut-point insertion for checking the equivalence high level software against RTL of combinational components. They introduce cut-points early during the analysis of the software model, rather than after generating a low level hardware equivalent. In this way, they overcome the exponential enumeration of software paths as well as the logic blow-up of tracking merged paths. However, it is necessary to synthesize word level information into bit level because they use BDD to represent the symbolic expressions and so the capacity is limited by memory and run time requirements. In addition the authors have only focused on combinational equivalence checking and have not addressed how to extend the proposed method for sequential equivalence checking purpose.

In all above approaches BDD or SAT based methods are utilized to represent symbolic expressions while algorithmic specifications such as those for digital signal processing contain a lot of arithmetic operations that should be encoded into bit level operations. Thus lower-level techniques like BDD or SAT are not able to handle these designs due to too large number of Boolean variables or clauses. On the contrary, in our previous work [3], we have presented a hybrid method of word-level solving for arithmetic parts and lower level SAT solving for Boolean parts that allowed us to automatically find equivalent nodes in the ASC and RTL. In this method a cut-plane is defined as a set of cut-points which should be specified in the two descriptions. At each plane, related symbolic expressions of the specification and implementation are constructed. After that a word level canonical representation called TED is used to avoid bit level analysis as much as possible due to logical complexity blow-up in the arithmetic expressions. However it is necessary to partition Boolean and arithmetic parts efficiently. This method also implies additional overhead because of switching between two engines.

3 LTED: Hybrid Representation

The goal of this section is to introduce a new graph-based representation called LTED for functions with a mixed Boolean and integer domain and an integer range to represent arithmetic operations at a high level of abstraction, while other proposed *Word Level Decision Diagrams* (WLDDs) are graph-based representations for functions with a Boolean domain and an integer range. In LTED, functions to be represented are maintained as a single graph in strongly canonical form. We assume that the set of variables is totally ordered and that all of the vertices constructed obey this ordering. Maintaining a canonical form requires obeying a set of conventions for vertex creation as well as weight manipulation. These conventions are similar to those of TED and are not discussed here for brevity [2].

In contrast to TED, LTED is a binary graph-based representation where the algebraic expression F(X,Y,…) is expressed by a first-order linearization of the Taylor series expansion [1]. Suppose variable *X* is the top variable of F(X,Y,…). Equation (1) shows F(X,Y,…), where *const* is independent of variable *X*, while *linear* is coefficient of variable *X*.

$$F(X, Y, \ldots) = const + X*linear. \quad (1)$$

LTED data structure consists of a *Variable (V)* node v that has as attributes an integer variable var(v) and two children const(v) and linear(v) which correspond to 0-child and 1-child in TED representation respectively. In order to normalize the

weights, any common factor is extracted by taking the greatest common divisor (gcd) of the argument weights. In addition, we adopt the convention that the sign of the extracted weight matches that of the *const* part (0-child). This assumes that gcd always returns a nonnegative value. Once the weights have been normalized the hash table is looked for an existing vertex or creates a new one. Similar to that of BDDs, each entry in the hash table is indexed by a key formed from the variable and the two children, i.e. *const* (0-child) and *linear* (1-child) parts. As long as all vertices are created, the graph will remain in strongly canonical form.

Example. Fig. 1 illustrates how the following algebraic expression is represented by LTED.

*f(X, Y, Z) = 24-8*Z+12*Y*Z-6*X*Y-6*X*Y*Z*

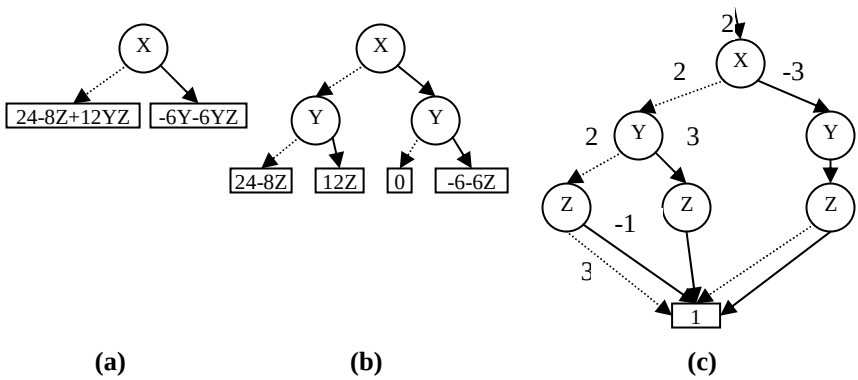


Fig. 1. LTED representation of 24-8Z+12YZ-6XY-6XYZ

Let the ordering of variables be X, Y, and Z. First the decomposition with respect to variable X is taken into account. As shown in Fig. 1 (a), *const* and *linear* parts will be 24-8*Z+12*Y*Z and -6*Y-6*Y*Z respectively. After that, the decomposition is performed with respect to variable Y as illustrated in Fig. 1 (b). Finally the expressions are decomposed with respect to variable Z and a reduced diagram is depicted. In order to reduce the size of an LTED, redundant nodes are removed and isomorphic sub-graphs are merged as shown in Fig. 1 (c). Analogous to TED and *BMDs, LTED is a canonical representation. In this representation, dashed and solid lines indicate *const* (0-child) and *linear* (1-child) parts respectively (more information can be found in [1]).

4 Application to Sequential Equivalence Checking

In this section, we explain a flow whereby an ASC is used to formally verify RTL using our proposed integrated graph-based representation that supports bit-level and word-level variables together. In order to improve our previous equivalence checking algorithm in [3], in this paper we introduce a modified equivalence checking approach.

Fig. 2 depicts modified equivalence checking algorithm. The inputs to the algorithm are a set of cut-planes (C) and an algorithmic specification in C (ASC) as well as an RTL description in Verilog (RTL). A cut-plane is a set of variables that are

interesting for observation in the two models. It should be noted that determining cut-planes is not a difficult task due to modular description of C code and hierarchical design of RTL code. Obviously, it is only necessary to determine some cut-planes in the ASC rather than specifying cut-planes [3] or corresponding of variables [5] in the two descriptions.

As shown in Fig. 2, first of all a cut-plane is chosen. The algorithm selects the nearest cut-plane to the primary inputs for better performance. This process can be done automatically by sorting cut-planes from primary inputs to primary outputs in the ASC. The selected cut-plane (CP) is removed from set of cut-planes (C) and LTED representation for all variables in the CP are constructed (ASC_LTED).

On the other hand, RTL description is synthesized using synthesis tools and modeled by a finite state machine with datapath (FSMD). The FSMD adds a datapath including variables and operators on communication to the classic FSM. The FSMD is represented as a transition table, where we assume each transition is executed in a single clock cycle. Operations associated with each transition of this model are executed in a sequential flow. Each controller transition is defined by the current state,

```
Sequential_EC (ASC: Algorithmic Level Model;
               RTL: RTL Model; C: set of Cut-planes)
WHILE (C is not empty)
     Select a cut-plane CP from C;
     C = C – CP;
     ASC_LTED = Generate LTED representation of all
                variables in CP based on their
               descriptions in ASC;
     WHILE (CP is not empty)
        RTL_LTED (t) = Generate LTED representations of
                       all variables are assigned to at
                       the current cycle (t) of RTL;
         IF v (a set of variables) ⊆ CP are assigned to
                            at the current cycle of RTL
          CP = CP – v;
          RTL_LTED_v = Get LTED representation of v
                       according to RTL_LTED (t);
          IF RTL_LTED_v is equivalent to some nodes in
                ASC_LTED
             Introduce primary inputs at v and related
             equivalent nodes in ASC model;
          Proceed to the next cycle;
```

Fig. 2. Sequential equivalence checking algorithm

the condition to be satisfied and a set of operations or actions. The condition evaluated true will determine the transition to be done and thus the actions to be executed. In other words, RTL model in Fig. 2 represents RTL description in a manner so that symbolic expressions assigned to each state or output variable can easily be extracted. After constructing ASC_LTED, the FSMD is traversed and after every cycle if some variables in CP are assigned to at the current cycle, appropriate symbolic expressions

are extracted (RTL_LTED_v) and during representing them by LTED, equivalent nodes will be found automatically due to canonical form of LTED representation. If we find some nodes that are equal to some points in the ASC_LTED, we can cut out the equivalent part and introduce new primary inputs in their places. These primary inputs are used while next iteration of *outer while loop* is executed. In the *inner while loop*, the algorithm proceeds to the next state of RTL model until all variables in the selected cut-plane are checked their equivalence with some nodes in the ASC. However, in the *outer while loop*, the process repeats until no cut-plane is available. If we can carry on this process to the outputs of the two descriptions, then we have formally verified equivalence.

In our previous work [3], during TED representation of the two descriptions, Boolean and arithmetic expressions need to be extracted separately and SAT solver was used to handle Boolean expressions. In contrast, as illustrated in Fig. 3, LTED is able to represent both Boolean and arithmetic expressions together and there is no need to use BDD or SAT solvers to handle Boolean expressions. In addition, we will be able to use word level information as much as possible and in contrast to [5], we have no need to encode arithmetic expressions into bit levels. If bit-slicing of some word-level variables (see *Var* as a word-level variable and *Var[j]* as a bit-slice of *Var* in Fig. 3) are used in Boolean part, in contrast to [3], the interaction between two parts are handled internally and decomposing result is represented by LTED. As shown in a gray box in Fig. 3, decomposition is performed into LTED representation so that whenever a decision is made in Boolean part, the appropriate value which can be a symbolic Boolean value *x* as illustrated in Fig. 3, is feedback to the arithmetic part and the related variable, i.e., *Var*, is decomposed into other word-level variables according to the following equation where Var_H and Var_L are new integer variables and *x* is a new Boolean variable:

$$Var = 2^{(j+1)} * Var_H + 2^j * x + Var_L$$

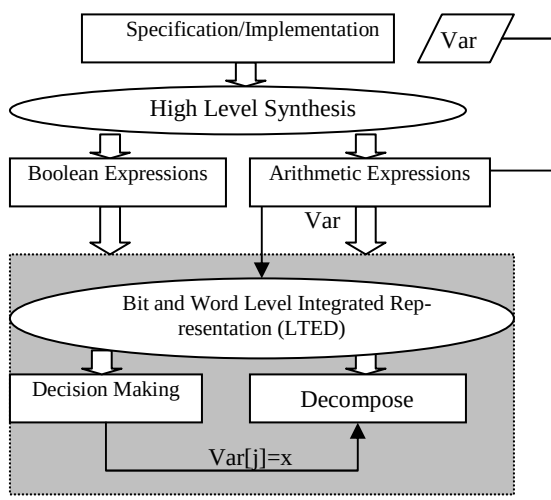


Fig. 3. Integrated equivalence checking approach

Table 1. IDCT benchmark with 2-positions and 4-positions decomposition

Cases			*case0*	*case1*	*case2*	*case3*	*case4*	*case5*	*case6*	*case7*
NN	*2-pos*		6072	8120	10184	12232	14320	16368	18432	20480
	4-pos		8120	12232	16368	20480	24560	28656	32768	36864
Var	*2-pos*		96	128	160	192	224	256	288	320
	4-pos		128	192	256	320	384	448	512	576
#add	*2-pos*		696	760	824	888	952	1016	1080	1144
	4-pos		760	888	1016	1144	1272	1400	1528	1656
#sub	*2-pos*		432	432	432	432	432	432	432	432
	4-pos		432	432	432	432	432	432	432	432
#mul	*2-pos*		624	688	752	816	880	944	1008	1072
	4-pos		688	816	944	1072	1200	1328	1456	1584
Mem	*2-pos*	*LTED*	7	9	12.2	14.4	16.7	18.6	21	24
		TED	12.1	14	15.8	18.1	20.3	25	28.3	timeout
	4-pos	*LTED*	9	14.4	18.6	24	30.2	37	44	50
		TED	14	18.1	25	timeout	timeout	timeout	timeout	timeout
Time	*2-pos*	*LTED*	1.6	2.2	2.9	3.5	4	4.6	5	5.8
		TED	185	365	614	972	1250	1985	2520	timeout
	4-pos	*LTED*	2.2	3.5	4.6	5.8	7.5	8.8	10	11.8
		TED	370	970	1990	timeout	timeout	timeout	timeout	timeout

5 Experimental Results

In the following we present some experiments with variable decomposition; in order to demonstrate that LTED can be used as an integrated canonical representation. LTED package was implemented in C++ and run on an Intel 2.1GHz Core Duo and 1 GByte of main memory running Windows XP. The experimental results will be compared to those of TEDify package which is a graph-based word level representation [2]. We have taken into account IDCT which has 690 lines C-code, 64 integer inputs and 64 integer outputs. It contains three loops which are iterated 64 and 8 times and in each iteration, 44 additions, 31 subtractions, 22 multiplications and 30 shift operations are computed on 16-bits operands which are symbolic variables. The RTL code has 984 lines which consist of 197 flip-flops and 2971 gates.

We follow up on comparing LTED and TED by employing the data reported in our previous work [3]. We have applied our approach on *IDCT* benchmark and divided experimental results into two configurations as tabulated in Table 1: (1) bit-slicing of a set of variables at 5^{th} and 10^{th} bit positions and (2) bit-slicing of a set of variables at 5^{th}, 10^{th}, 15^{th}, and 20^{th} bit positions. Eight cases of a set of variables are taken into account to apply various bit-slicing to all variables in *IDCT* test case. These cases are described as follows:

$case_0$: $b_{(8j)}$; for j=0 to 7
$case_i$: $b_{(i+8j)10}$ *and* $case_{i-1}$; for j=0 to 7 and i=1 to 7

For instance $case_0$ only comprises b_0, b_8, b_{16}, b_{24}, b_{32}, b_{40}, b_{48} and b_{56} variables while $case_7$ consists of all 64 variables. For *2-pos*, after increasing the number of decomposed variables to 64 variables (column *case7* in Table 1), TED package failed while it spent more than 40 minutes CPU time. In addition, for *4-pos*, after increasing the number of variables to be decomposed to 32 variables (column *case3* in Table 1), TED package could not handle this case and failed after spending 40 minutes. Consider *case2* where 24 variables should be decomposed. Column *case2* in Table IV indicates the number of nodes (*NN*) and the number of input variables (*Var*) after decomposing all 64 input variables according to different bit positions (*2-pos* and *4-pos*). The numbers of input variables for *2-pos* and *4-pos* decomposition are 160 and 256 respectively. Although the number of nodes increases from 10184 to 16368, memory needs in LTED package grow from 12.2 MB to 18.6 MB. Furthermore, run time required to check the equivalence between two descriptions increases from 2.9 seconds to 4.6 second for LTED package, while TED package requires 614 seconds for *2-pos* and 1990 seconds for *4-pos* which are almost two orders of magnitude larger than those of LTED package.

References

1. Alizadeh, B., Navabi, Z.: Word level symbolic simulation in processor verification. IEE Proceedings Computers and Digital Techniques Journal 151(5), 356–366 (2004)
2. Ciesielski, M., Kalla, P., Askar, S.: Taylor Expansion Diagrams: A Canonical Representation for verification of data flow designs. IEEE Transactions on computers 55(9), 1188–1201 (2006)
3. Alizadeh, B., Fujita, M.: A hybrid approach for equivalence checking between system level and RTL descriptions. In: 6th International Workshop on Logic and Synthesis (IWLS), pp. 298–304 (2007)
4. Koelbl, A., Lu, Y., Mathur, A.: Embedded tutorial: Formal equivalence checking between system-level models and RTL. In: Proceedings of ICCAD, pp. 965–971 (2005)
5. Vasudevan, S., Viswanath, V., Abraham, J., Tu, J.: Automatic Decomposition for Sequential Equivalence Checking of System Level and RTL Descriptions. In: Proceedings of Formal Methods and Modes for Co-Design (Memocode), pp. 71–80 (2006)
6. Feng, X., Hu, A.: Early Cutpoint Insertion for High-Level Software vs. RTL Formal Combinational Equivalence Verification. In: Proceedings of the 43th Design Automation Conference (DAC), pp. 1063–1068 (2006)
7. Kroening, D., Clarke, E., Yorav, K.: Behavioral consistency of C and Verilog programs using bounded model checking. In: Proceedings of the 40th Design Automation Conference (DAC), pp. 368–371 (2003)
8. Jain, H., Kroening, D., Clarke, E.: Verification of SpecC using Predicate Abstraction. In: Proceedings of Formal Methods and Models for Co-Design (Memocode), pp. 7–16 (2004)
9. Karfa, C., Mandal, C., Sarkar, D., Pentakota, S.R., Reade, C.: A Formal Verification Method of Scheduling in High-level Synthesis. In: Proceedings of the 7th International Symposium on Quality Electronic Design (ISQED), pp. 71–78 (2006)

A Circuit Model for Fault Tolerance in the Reliable Assembly of Nano-systems

Masoud Hashempour, Zahra Mashreghian Arani, and Fabrizio Lombardi

Northeastern University, Boston, MA, 02115, USA
Tel: 617-373-4854, Fax.: 617-373-8970
{masoud,zahra,lombardi}@ece.neu.edu

Abstract. As still in infancy, emerging technologies are exhibiting high defect rates; this is mostly due to the stochastic nature of the bottom-up chemical and physical self-assembly processes that are commonly employed in manufacturing. As relationships between components are very complex, a system-level solution is commonly pursued. This challenge is commonly identified with the generic problem of building reliable systems out of unreliable components. In this paper, a novel model which exploits the universality of some circuits (such as a multiplexer), is proposed. In this model a feature that is employed in the presence of errors due to faulty gates, is given by the capability of a restorative stage to have a controlled functional relationship between inputs and the output. Using bifurcation and its geometric representation, the gate error probability is analyzed in detail. Differently from a single gate, the 4-to-1 multiplexer based implementation is considered as instance of a multi-level universal (MLU) circuit. It is proved that in the presence of multiple faulty gates, compensation takes place among them, such that a correct output is still generated at a higher threshold failure probability (for a region bounded within the values of 0.471 and 0.523) as compared with previous schemes. Due to its MLU nature, the proposed model operates in a mode that allows generality and flexibility in implementation for threshold analysis.

Index Terms: fault-tolerance, nanotechnology, bifurcation, multiplexing, circuit model.

1 Introduction

The fabrication and manufacturing of submicron and nanometer-scale devices are challenging processes; sensitivity to external factors (such as cosmic radiation, electromagnetic migration, interference, and thermal fluctuations) results in the almost unavoidable occcurrence of permanent faults (due to defects during the manufacturing process) and transient faults (during the operational life of the system). *Tolerance* to these faults is required at all levels of design and integration. For emerging technologies (such as Quantum-dot Cellular Automata (QCA) [7]), defects have been reported at manufacturing due to the unique features of the employed processes (such as for biological and molecular implementations) [8].While device- and circuit-level solutions are sought in the initial

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 705–713, 2008.

stage of technology development (to assess process monitoring and scaling), it is anticipated that *system-level solutions* will be ultimately required to cope with the complexity of these systems and the difficulty to properly model interactions at lower levels. The seed manuscript of [1] has proposed the so-called *NAND multiplexing* technique as a possible solution to achieve a reliable system assembly from unreliable components; this scheme was originally developed to circumvent the poor reliability of tubes for the assembly of digital computers in the early 1950s. For fault tolerance, [1] utilizes a high degree of redundancy through the use of two types of circuits (majority voters and NAND gates for restoration) under the assumption that circuits are made of not fully reliable components. [1] has proved that if the failure probability of the gates in the components is sufficiently small and statistically independence is assumed for failures, then computation can be reliably performed with high probability. A similar scenario is also applicable to nanotechnology [8]. High defect rates have been experienced for components at submicron and nano ranges, especially in molecular implementations. Moreover as a high probability of failure of components is encountered, new fault tolerant techniques must employ redundancy for assembling systems which are expected to have an extremely large number of components. Nanotechnology components are *inherently unreliable* [8] and corrective actions (such as restoration and in-situ correction and detection of defects and faults) are required because due to the high density and large number of defective components, it is not possible to replace or isolate them. As a *restorative stage* for fault tolerance, NAND multiplexing has been analyzed in the literature [6] [9]; bounds for the maximum fault probability of each device and the reliability of NAND multiplexing have been treated extensively [9]. In 1977, [2] presented a rigorous proof to improve von Neumann's result by showing that logarithmic redundancy is sufficient for any boolean function. [3] has shown that for so-called "noisy" NAND gates, the maximum probability of failure (also known as threshold) for each component is $\frac{(3-\sqrt{7})}{4} \approx 0.08856$. Recently, [4] has established through a reliability analysis of NAND multiplexing that the use of additional restorative stages improves performance. [5] has considered the use of restorative stages to improve system reliability; at small fault rates, an increase in the number of restorative stages improves reliability, while at high fault rates the increase in restorative stages may result in a degradation of the reliability [5]. Also [5] has shown that the analysis of [4] is only partially correct, By considering a different model for the so-called "U" random permutation, it has been shown that the results of [4] are not always the upper or lower reliability bounds. The objective of this paper is to propose a new implementation for the restorative stage in assemblying nano-systems made of highly unreliable components. This arrangement is based on a novel model by which multi-level universal (MLU) operation occurs in the circuit. Dynamic operation refers to the capability of *selectively adjusting* the functional relation between the inputs and the output, such that through a *multi-level implementation*, multiple failures can be tolerated in the gates, while still providing on a probabilistic basis the correct value at the output. This effect is referred to as *compensation* and in

(a) (b)

Fig. 1. (a) Generic Dynamic Circuit Model, (b) Multiplexer Implementation of NAND under the Generic Dynamic Model

this paper, it will be proved that in a MLU circuit, the probability of compensation occurrence depends *only* on the number of gates, not on the function of the model and the input value. A multiplexer (in a 4-to-1 arrangement) is proposed for *implementation.* The compensation provided by this multiplexer implementation is extremely close to the theoretical limit, thus providing *near optimality* for application to nano-systems at an increased threshold probability in gate failure (for a region bounded by ϵ_*=0.471 and 0.523). The paper is organized as follows. In Section 2, a Generic Dynamic Model is proposed; as implementation, a 4-to-1 multiplexer is treated in detail. Section 3 presents a detailed analysis of error compensation. Section 4 extends the bifurcation analysis to the multiplexer implementation. The last Section presents some very important observations as result of the presented anlaysis and concludes this paper.

2 Generic Dynamic Model

In this section, a new generic model for assemblying reliable systems from unreliable components is proposed. This model utilizes as implementation the universal nature of a multiplexer and its flexibility in generating different circuits inclusive of NAND. In the proposed model, the so-called *MLU feature* of a circuit is utilized. A circuit is said to be MLU provided if it has the feature of changing its output using control signals (as selectors). Fig. 1(a) shows a MLU circuit model that satisfies the previous definition. This circuit has four input functions and single output; few control lines (as selectors) are also provided. As an example of a nanotechnology, a QCA implementation of this circuit would require fixed polarity cells for the Fi signals, so only the two input selector lines (as in a NAND gate) would be required [7]. Without loss of generality and correctness, the model can be extended to m input lines with $log_2 m$ control lines per output. Examples of circuit that exhibit MLU behavior in implementation, are memories (such as look-up-tables) and multiplexers. As an example, assume only a single NAND gate (as in previous papers [6]) is utilized as a building block of a probabilistic structure for restoration. As a single gate, a NAND is static because no additional controllability can be exercised over the inputs. This is not applicable to a MLU circuit; under the proposed MLU model, the following parameters can be defined for controllability:

i. The probability of the two input signals as selectors is given by

$$P^1_{In_1} = X, \quad P^0_{In_1} = 1 - X, \quad P^1_{In_2} = Y, \quad P^0_{In_2} = 1 - Y \tag{1}$$

ii. The probability of gate error (for both cases of faulty and error free, also referred to as *perfect*) is given as follows

$$P_{gate}(faulty) = \epsilon, \quad P_{gate}(error\ free\ \ or\ \ perfect) = 1 - \epsilon \tag{2}$$

Next, consider a two-input NAND gate implemented as a MLU circuit under the proposed model. The multiplexer is a universal circuit that can implement any combinational function, so it is also possible to implement the functionality of the NAND gate through the proposed MLU circuit of Fig. 1(a) as a 4-to-1 multiplexer (Fig. 1(b)). As shown in Fig. 2(a), the multiplexer implementation under this model has three gate levels(i. One output OR gate, ii. Four AND gates, and iii. Two Inverters). By comparing the multiplexer implementation to the generic MLU circuit of Fig. 1(a), the following conditions are applicable (the gate-level structure of the proposed multiplexer implementation is depicted in Fig. 2(a)):

$$F_1 = F_2 = F_3 = \text{“1”}, \ F_4 = \text{“0”}, \ Selectors = In_1, In_2 \tag{3}$$

Assume that each faulty gate causes a bit flip in its output and $In_1 = In_2 =$“0”; in the error free circuit, the output is “1”. If A_1 is faulty, then the output will be faulty too (“0”); however if both A_1 and N_2 are faulty, then the output is error free , i.e. the combination of two faulty gates results in an error free output. This effect is referred in this paper as *compensation*, i.e. for the above example, a fault in N_2 compensates the fault in A_1. This is a feature of MLU circuits that is caused by the multi-level implementation in the proposed model (assuming independence in the gates' failures). This behavior however, does not occur in a NAND gate as a single device, hence static structures (such as a single NAND gate) have no compensation. Such feature can be analyzed on a probabilistic basis; for a static circuit the error compensation probability is 0, while its implementation as a MLU circuit results in a non zero probability. As a large number of faulty gates is present in nano scale systems [8], then it is realistic to expect that compensation in MLU circuits can be used for designing more reliable circuits. This aspect will be rigourously analyzed next.

3 Error Compensation

With no redundancy, it will be shown that the error compensation of a MLU circuit depends on the number of gates. The circuit is defined by the following parameters:

i. n denotes the number of gates in the circuit.
ii. i is the number of faulty gates in the circuit.
iii. $Err(i)$ is the number of states for which no compensation occurs in the presence of i faulty gates (or errors).

iv. $Comp(i)$ is the number of states for which compensation occurs in the presence of i faulty gates (or compensations).

Therefore for each i,

$$Err(i) + Comp(i) = \binom{n}{i} \tag{4}$$

For all states,

$$\sum_{i=1}^{n} Err(i) + \sum_{i=1}^{n} Comp(i) = \sum_{i=1}^{n} \binom{n}{i} = 2^n - 1 \tag{5}$$

Also in the proposed model and implementation, four distinct states are possible in the system; these states are given as follows:

i. **All gates are fault free (perfect):** no compensation and no error are present.
ii. **All gates are perfect except the output gate (OR gate):** an error will be present at the output of the circuit.
iii. **All other states in which the output gate is perfect:** $\frac{2^n-2}{2}$ states exist. In this case, assume that the total number of errors is equal to K and the total number of compensations is given by L, so

$$\sum_{states} Err(i) = K, \quad \sum_{states} Comp(i) = L, \quad K + L = \frac{2^n - 2}{2} \tag{6}$$

iv. **All other states in which the output gate is faulty:** again $\frac{2^n-2}{2}$ states are possible. In this case the same scenario as in iii above is applicable, but there is an inversion at the output (because the output gate is faulty). Therefore, the total number of errors is equal to the total number of compensations in the previous state (L) and the total number of compensations is equal to the total number of errors in the previous state (K).

So, the total numbers of errors and compensations for all states are as follows:

$$\sum_{i=1}^{n} Err(i) = K + L + 1 = 2^{n-1}, \quad \sum_{i=1}^{n} Comp(i) = K + L = 2^{n-1} - 1 \tag{7}$$

In percentage, compensation of the proposed implementation of the MLU model is given by

$$Compensation\% = \frac{\sum_{i=1}^{n} Comp(i)}{\sum_{i=1}^{n} Comp(i) + \sum_{i=1}^{n} Err(i)} = \frac{2^{n-1} - 1}{2^n - 1} \tag{8}$$

The above analysis shows that compensation depends only to the number of gates, not on the function of the model or the inputs value. For a very large number of gates,

$$\lim_{n \to \infty} \left(\frac{2^{n-1} - 1}{2^n - 1} \right) = \frac{1}{2} = 50\% \tag{9}$$

In the proposed 4-to-1 multiplexer based implementation of a NAND, $n = 7$, so its compensation is given by

$$compensation = \frac{2^6 - 1}{2^7 - 1} = \frac{63}{127} = 49.606299\% \tag{10}$$

For n=1 (a single NAND gate), the compensation is 0, thus proving the limitations of existing NAND multiplexing arrangements. The above analysis proves also that for the proposed NAND MLU implementation (using a 4-to-1 multiplexer as an instance of a circuit), compensation is very close to the maximum theoretical limit.

4 Bifurcation of Multiplexer-Based NAND

As proved in previous sections, compensation is a powerful feature to consider when assemblying nano-systems using unreliable components. The multiplexer implementation as instance of a MLU circuit has a compensation very near to its theoretical limit; this feature will be further analyzed in the binary tree structure by replacing the NAND gates with the multiplexer implementation of a MLU circuit; bifurcation is used to generate its map. In the multiplexer structure, it is assumed (as commonly found in existing literature [6]) that all gates have the same Gate Error Probability or GEP (ϵ) and all connections are fault free (perfect). For calculating the probability of having "1" at the output, all cases must be considered. four states are possible for the values of the inputs of the two control lines under the exhaustive combination of the status of the two inverters. Because of the lack of space only the end results have been shown.

4.1 In_1=In_2=1

When the two control input lines are in the "1" state, the probability of having "1" at the output can be calculated as follow:

$$P^1_{In_1=In_2=1} = 4\epsilon^7 - 28\epsilon^6 + 67\epsilon^5 - 83\epsilon^4 + 62\epsilon^3 - 29\epsilon^2 + 7\epsilon \tag{11}$$

4.2 In_1=In_2=0

The equation for total probability of the output being "1" for the case of $In_1 = In_2 = 0$, is given by

$$P^1_{In_1=In_2=0} = -4\epsilon^7 + 24\epsilon^6 - 59\epsilon^5 + 76\epsilon^4 - 54\epsilon^3 + 20\epsilon^2 - 4\epsilon + 1 \tag{12}$$

4.3 $In_1 = 0, In_2 = 1$ or $In_1 = 1, In_2 = 0$

The following equation is obtained for the probability of the output being "1" for the case that $In_1 = 0, In_2 = 1$ or $In_1 = 1, In_2 = 0$:

$$P^1_{In_1=1,In_2=0} = P^1_{In_1=0,In_2=1} = -4\epsilon^7 + 20\epsilon^6 - 43\epsilon^5 + 51\epsilon^4 - 35\epsilon^3 + 13\epsilon^2 - 3\epsilon + 1$$

Therefore, it is now possible to establish the probability of having "1" at the output under the new proposed model for the 4-to-1 implementation as

$$Z = f(X,Y) = XYP^1_{In_1=In_2=1} + (1-X)(1-Y)P^1_{In_1=In_2=0} + (1-X)YP^1_{In_1=0,In_2=1} + X(1-Y)P^1_{In_1=1,In_2=0} \quad (13)$$

With the same probability for having "1" at the gate inputs ($X = Y$) and also $P^1_{In_1=1,In_2=0} = P^1_{In_1=0,In_2=1}$, then

$$Z = X_{n+1} = f(X_n) = (P^1_{In_1=In_2=0} + P^1_{In_1=In_2=1} - 2P^1_{In_1=1,In_2=0})X_n^2 + 2(P^1_{In_1=1,In_2=0} - P^1_{In_1=In_2=0})X_n + P^1_{In_1=In_2=0} \quad (14)$$

The derivative of this function is :

$$f'(X_{n+1}) = 2(P^1_{In_1=In_2=0} + P^1_{In_1=In_2=1} - 2P^1_{In_1=1,In_2=0})X_n + 2(P^1_{In_1=1,In_2=0} - P^1_{In_1=In_2=0}) \quad (15)$$

$f'(x)$, the fixed points and the bifurcation map of the multiplexer implementation of the proposed MLU model are similar to the single NAND Gate but the most interesting feature of this model is that the above formulation of this model is really close to a fractal function. Precisely if we had $9\epsilon^2$ instead of $13\epsilon^2$ in $In_1 = 0, In_2 = 1$ (or $In_1 = 1, In_2 = 0$) cases the end result was a fractal function which its bifurcation map is plotted in Fig. 2(b). Fig. 2(c) shows the enlarged region for the fractal function to allow a reliable system assembly using unreliable components (prior to the fractal/chaotic region occuring at values higher than 0.523 for ϵ_*). The bifurcation map of this fractal function has four regions in the range of 0 to 1 for ϵ_*.

1. The first region is periodic, but its error probability (ϵ) is very close to zero, so it has very limited usefullness.
2. The second region is not periodic (fixed points).
3. The third region is also a periodic region centered at ϵ_*=0.5, and can be used for operation in assemblying reliable systems out of unreliable components. This region is bounded by 0.471 and 0.523 as the value for ϵ_* prior to the fractal region.
4. The fourth region is the so-called chaos region and therefore, it can not be used.

So by using a similar model for molecular electronic devices with probablistic output, it's possible to achive a MLU with fractalic behaviour. This makes the proposed implementation very attractive for nanotechnology. Finding a specific molecular device is under investigation and the results will be presented in future papers. Consider the model of [6]. A binary tree made of NAND/NOR gates is a discrete MLU system in which each element (NAND/NOR) on the tree has a static behavior in the presence of an error in the gate. The bifurcation map for these two binary trees

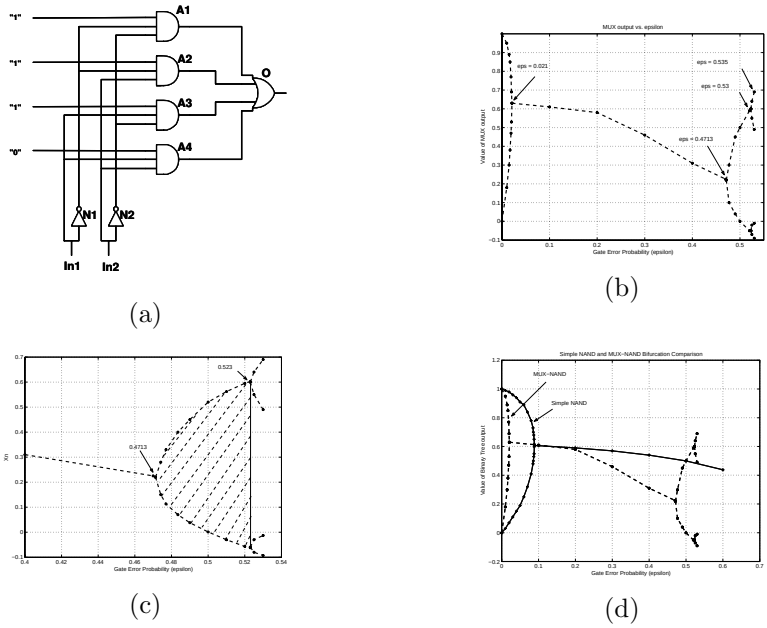


Fig. 2. (a) Gate-Level Implementation of the Multiplexer-based NAND under the Generic Dynamic Model, (b) Bifurcation Map Diagram for Fractal Function , (c) Enlarged Periodic Region for Fractal Function, (d) Simple-NAND and MUX-NAND Bifurcation Comparison

shows that a periodic region occurs only near $\epsilon_* = 0$ while a long non-periodic region (fixed point) occurs over the remaining range (up to 1) of ϵ_*. If the static gate element of such binary trees is changed into the multiplexer-based MLU implementation of the NAND/NOR gate, error compensation occurs with beneficial consequences as reflected in a new bifurcation map.

5 Discussion and Conclusion

The model and implementation proposed in this paper can be compared with a multiplexing scheme using single NANDs based on different features as figures of merit.

i. The derivative of the multiplexer-based implementation has two points for which $|f^{'}(x)| = 1$, so more than a single area have the feature of a periodic function compared with the single NAND scheme for NAND multiplexing. It is also worth to note the fast slope of the plot for the multiplexer-based implementation of the NAND.
ii. Figure 2(d) shows the bifurcation map for the two implementations. It is significant to point out that for ϵ_*=0.4713, (up to ϵ_*=0.523) reliable system assembly is now possible using the proposed implementation. These are

significantly higher values of thereshold failure probability compared with existing techniques (bounded at ϵ_*=0.088).

These features basically prove that the multiplexer-implementation of the NAND as an instance of a MLU circuit under the proposed model results in a significant improvement over previous arrangements (mostly single NAND based) for the restorative stage. As a multi-level implementation of a MLU circuit (such as the multiplexer) result in a higher overhead, the substantial increase in threshold failure probability (due to compensation in the extremely huge number of components)is a very positive feature for assembly nano-based systems using emerging technologies [8]. The analysis reported in this paper has shown that if each gate fails independently and unbiased in a bistable fashion (such as by tossing a coin in probability theory), then over a long run (i.e. for large values of n) the assembly of such a system will be probabilistically possible using restorative stages implemented by multiplexer-based NANDs. Moreover due to the large density of emerging technologies, the increase in complexity of the restorative stage can be easily accomodated (albeit a two-level implementation introduces a longer delay). The 4-to-1 multiplexer implementation presented in this paper is very close to the theoretical limit by which compensation due to multiple faults can probabilistically result in a correct output by a restorative stage. The use of the proposed implementation for QCA in molecular implementations is currently being evaluated to reach a fractalic behaviour.

References

1. von Neumann, J.: Probabilistic logics and the synthesis of reliale organisms from unreliable components. In: Shannon, C.E., McCarthy, J. (eds.) Automata Studies, pp. 43–98. Princeton University Press, Princeton (1955)
2. Dobrushin, R.L., Ortyukov, S.I.: Upper bound on the redundancy of self-correcting arrangements of unreliable functional elements. Prob. Inform. Trans. 13, 203–218 (1977)
3. Evans, W., Pippenger, N.: On the maximum tolerable noise for reliable computation by formulas. IEEE Transactions on Information Theory 44, 1299–1305 (1998)
4. Han, J., Jonker, P.: A system architecture solution for unreliable nonoelectronic devices. IEEE Trans. on Nanotechnology 1(4), 201–208 (2002)
5. Norman, G., Parker, D., Kwiatowska, M., Shukla, S.K.: Evaluating the reliability of defect-tolerant architectures for nanotechnology with probabilistic model checking. In: Proceedings of the 17th International Conference on VLSI Design (VLSID 2004), Mumbai, India, pp. 907–912. IEEE Computer Society, Los Alamitos (2004)
6. Qi, Y., Gao, J., Fortes, J.A.B.: Probabilistic computation: a general framework for fault tolerant nanoelectronic systems. Adv. Comput. Inform. Processing Lab., Univ. Florida, Gainesville, FL, Tech. Rep. TR-ACIS-03-002 (November 2003)
7. Lent, C.S., Tougaw, P.D.: A device architecture for computing with quantum dots. Proc. of the IEEE 85, 541–557 (1997)
8. Han, J., Taylor, E., Gao, J., Fortes, J.: Reliability modeling of nanoelectronic circuits. In: Proc. 5th IEEE Conference on Nanotechnology, Nagoya (2005)
9. Gao, J.B., Qi, Y., Fortes, J.: Bifurcations and fundamental error bounds for fault-tolerant computations. IEEE Trans. on Nanotechnology (to appear)

An Operator for Removal of Subsumed Clauses

Mohammad Ghasemzadeh and Christoph Meinel

Yazd University and HPI at the University of Potsdam
m.ghasemzadeh@yazduni.ac.ir, meinel@hpi.uni-potsdam.de

Abstract. The formal verification problem is strongly correlated with the SAT and the QSAT problems. One approach in solving SAT and QSAT problems is based on using ZDD (Zero-Suppressed BDD) which is a variant of BDD (Binary Decision Diagram). When working with clauses, very often a big number of clauses are subsumed by other clauses. In such situation one may remove the subsumed clauses to simplify the formula. In this paper we reintroduce an operator for removing subsumed clauses, then we show how removing subsumed clauses may affect the efficiency of ZUNFOLD which is our 'Unfolding' based QSAT solver.

Keywords: Subsumed clauses, Formal verification, Zero-Suppressed Binary Decision Diagram (ZDD), Quantified Boolean Formula (QBF), Satisfiability.

1 Introduction

During last few your many researchers have worked on using propositional logic as a framework for knowledge representation and problem solving. Propositional satisfiability (SAT) is a central problem in computer science with numerous applications. Satisfiability of quantified Boolean formula-QBF, also known as QSAT is a generalizetion of the SAT problem. Many computational problems such as constraint satisfaction problem, some problems in graph theory, planning and verification can be formulated easily as instances of SAT or QSAT problems [1].

Ordered binary decision diagram [2,3] and its variants are data structures which have shown to be very useful in implementation of verification tools. While BDDs are better suited for representing Boolean functions, ZDDs are better for representing sets. If we consider the variables appearing in a QBF as a set then the matrix of the QBF (which is a Boolean formula in CNF) can be viewed as a set of subsets. This is the motivation for using ZDD to represent and process QBF formula. This idea has already been used in a number of related works [4,5].

In ZUNFOLD [6] we adopted and implemented the unfolding algorithm to be suitable for working with ZDD. During evaluation of our algorithm we realized that the unfolding algorithm would not work well if we wouldn't equip it with remove-subsumed operation. In this paper we reintroduce ZUNFOLD then we discuss about subsumed clauses, a formal definition for removing subsumed clauses when presented by means of ZDD and our CUDD based implementation.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 714–717, 2008.

2 Preliminary

2.1 Encoding Boolean Functions with BDD/ZDD

A BDD or more precisely ROBDD is a directed acyclic graph with labeled nodes, a unique source node, and such that each node is either a sink node or an internal node. An internal node can be denoted by $\Delta(x, n_1, n_2)$ where x is the label and n_1, n_2 stand for its two children [7]. The main advantage of using BDD / ZDD for encoding a propositional formula is that it can describe very large sets of models in a very compact way. However this requires the computation of the Shannon normal form of the formula, which could be very expensive.

2.2 Subsumed Clauses

In SAT and QSAT problems when we are working with CNF formulas, in order to prune the search space, variables which can get only one value must be assigned and be removed from the formula. A *unit clause* is a clause with exactly one literal. For example in $f=(a \vee \neg b \vee \neg c) \wedge (a \vee \neg c \vee d) \wedge (b) \wedge (a \vee b \vee c)$, the third clause is a unit clause. Since the algorithm tries to satisfy the formula, the variable appearing in a unit clause can essentially get one of the truth-values. In the above example, b can only receive the value *true*, which lets f be simplified to: $f_1=(a \vee \neg c) \wedge (a \vee \neg c \vee d)$. In f_1 the first clause *subsumes* the second clause. In other words, if an assignment satisfies the first clause then it will surely satisfy the second clause too. In such cases the subsumed clause(s) can be removed.

In order to remove such subsumed clauses we need to define an operator for subtraction of the subsumed clauses. A formulation for this kind of subsumed difference can be seen in Fig. 1.

$$
\begin{aligned}
&T_1: 0 \setminus\setminus A = 0 \\
&T_{2a}: 1 \setminus\setminus 1 = 0 \qquad T_{2b}: \text{if } A \neq 1, 1 \setminus\setminus A = 1 \\
&T_{3a}: \Delta(l, A_1, A_2) \setminus\setminus 0 = \Delta(l, A_1, A_2) \\
&T_{3b}: \Delta(l, A_1, A_2) \setminus\setminus 1 = 0 \\
&R_1: (l > m)\ \Delta(l, A_1, A_2) \setminus\setminus \Delta(m, B_1, B_2) = \\
&\qquad \Delta(l, A_1, A_2) \setminus\setminus B_2 \\
&R_2: (l < m)\ \Delta(l, A_1, A_2) \setminus\setminus \Delta(m, B_1, B_2) = \\
&\qquad \Delta(l,\ A_1 \setminus\setminus \Delta(m, B_1, B_2),\ A_2 \setminus\setminus \Delta(m, B_1, B_2)) \\
&R_3: \Delta(l, A_1, A_2) \setminus\setminus \Delta(l, B_1, B_2) = \\
&\qquad \Delta(l,\ (A_1 \setminus\setminus B_1) \setminus\setminus B_2,\ (A_2 \setminus\setminus B_2))
\end{aligned}
$$

Fig. 1. The Subsume difference operator (Adopted from [7])

We implemented the above operation in CUDD [8]. A part of our code can be seen below.

```
DdNode* SubsumedDifference( DdNode *A, DdNode *B )
 {
  DdNode *A1, *A2, *B1, *B2, *R1, *R2, *R;
  int    l, m;

  if ( A==Zero ) return(Zero);
  if ( A==One &&  B==One ) return(Zero);
  if ( A==One &&  B!=One ) return(One);
  if ( B==Zero ) return(A);
  if ( B==One  ) return(Zero);

  l=Cudd_NodeReadIndex(A);  m=Cudd_NodeReadIndex(B);
  A1 = Cudd_T(A);  A2 = Cudd_E(A);
  B1 = Cudd_T(B);  B2 = Cudd_E(B);

  if ( l>m  ) return( SubsumedDifference( A , B2 ) );

  if ( l<m  )
   {
     R1 = SubsumedDifference( A1 , B );
     R2 = SubsumedDifference( A2 , B );
     R  = cuddZddGetNode( manager , l , R1 , R2 );
     Cudd_Ref( R ); return( R );
   }

  // l==m
  R1 = SubsumedDifference( SubsumedDifference( A1 , B1 ) , B2 );
  R2 = SubsumedDifference( A2 , B2 );
  R  = cuddZddGetNode( manager , l , R1 , R2 );
  Cudd_Ref( R ); return( R );
 }
```

2.3 Quantified Boolean Formulas

Quantified Boolean formula (QBF) is an extension of propositional formula (also known as Boolean formula). A Boolean formula like $(x \vee (\neg y \rightarrow z))$ is a formula built up from Boolean variables and Boolean operators like conjunction, disjunction, and negation. In quantified Boolean formula, quantifiers may also appear in the formula, like in $\exists x\ (x \wedge \forall y\ (y \vee \neg z))$. The $\exists$ symbol is called existential quantifier and the $\forall$ symbol is called universal quantifier. ZUNFOLD [6] is our already published QBF evaluator which is based on ZDD and unfolding. We added the above operation to investigate its effects.

3 Experimental Results

We examined ZUNFOLD to find out the effect of including the NoSub operation. Therefore we evaluated it by different benchmarks from QBFLIB [9] website. In order to investigate the effect of including/excluding the NoSub operation, we run it with/without the operation. Table 1 show the results of this experiment over LET's benchmarks. The third column titled "Av(Ratio)" stands for "average of ratio". Here *ratio* is the size of the ZDD before and after applying the NoSub operation. We can see the ratios rise as the size of the benchmark problem rise. This indicates that NoSub helps more when instances of the QSAT go larger. The last column in the

Table 1. Effect of the NoSub operation on ZUNFOLD

Problem	ZUNFOLD	Av(HitRatio)	Max(HitRatio)	ZUNFOLD*
tree-exa-10-10	<.01	2.11	05	50
tree-exa-10-15	<.01	3.08	06	660
tree-exa-10-20	.01	3.22	06	NotSolved
tree-exa-10-25	.02	3.98	09	NotSolved
tree-exa-10-30	.04	4.33	09	NotSolved
tree-exa-2-10	<.01	4.61	11	NotSolved
tree-exa-2-15	<.01	4.76	12	NotSolved
tree-exa-2-20	<.01	4.88	12	NotSolved
tree-exa-2-25	<.01	5.01	14	NotSolved
tree-exa-2-30	<.01	5.25	19	NotSolved
tree-exa-2-35	<.01	5.28	21	NotSolved
tree-exa-2-40	<.01	5.53	24	NotSolved
tree-exa-2-45	<.01	5.72	27	NotSolved
tree-exa-2-50	<.01	5.87	31	NotSolved

table shows that ZUNFOLD could only solve very small instances of QSAT when the NoSub operation was excluded.

4 Conclusion

We showed that using ZDD along with removal of subsumed clauses let us to implement an unfolding based QBF evaluator (QSAT solver) that works efficiently even for large instances of the problem.

References

1. Egly, U., Eiter, T., Tompits, H., Woltran, S.: Solving Advanced Reasoning Tasks using Quantified Boolean Formulas. In: 17th National Conference on Artificial Intelligence (AAAI 2000), pp. 417–422. AAAI/MIT Press (2000)
2. Bryant, R.E.: Graph-based algorithms for Boolean function manipulation. IEEE Transactions on Computers C-35, 677–691 (1986)
3. Meinel, C., Theobald, T.: Algorithms and Data Structures in VLSI Design. Springer, Heidelberg (1998)
4. Aloul, F.A., Mneimneh, M.N., Sakallah, K.A.: Backtrack Search Using ZBDDs. In: Int. Workshop on Logic and Synthesis (IWLS), page 5. University of Michigan (June 2002)
5. Aloul, F.A., Mneimneh, M.N., Sakallah, K.A.: ZBDD-Based Backtrack Search SAT Solver. In: Int. Workshop on Logic and Synthesis (IWLS), New Orleans, Louisiana, pp. 131–136 (2002)
6. GhasemZadeh, M., Meinel, C.: Splitting Versus Unfolding. In: 7th International Symposium on Representations and Methodology of Future Computing Technology, Tokyo, Japan (2005)
7. Chatalic, P., Simon, L.: Multi-Resolution on Compressed Sets of Cluases. In: 12th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2000) (2000)
8. Somenzi, F.: CUDD: CU Decision Diagram Package, `ftp://vlsi.colorado.edu/pub/`
9. QBFLIB - QBF Satisfiability Library, `http://www.mrg.dist.unige.it/qube/qbflib/`

Low Power and Storage Efficient Parallel Lookup Engine Architecture for IP Packets

Alireza Mahini[1], Reza Berangi[2], Hossein Mohtashami, and Hamidreza Mahini

[1] Academic Staff Member of Computer Engineering Department
Islamic Azad University-Gorgan Branch, Iran
[2] Asistant Professor of Computer Engineering Department
Iran University of Science and Technology, Iran

Abstract. Ternary Content-Addressable Memories are becoming very popular for designing high-throughput address lookup engines on routers: they are fast, cost-effective and simple to manage. Despite the TCAMs speed, their high power consumption is their major drawback.In this document, we designed a heuristic to match entries in TCAM stages so that only a bounded number of entries are looked up during the search operation.

The performance evaluation of the proposed approach shows that it can save considerable amount of routing table's power consumption.

Keywords: TCAM, Router, LPM And IP lookup.

1 Introduction

Forwarding of Internet Protocol (IP) packets is the primary purpose of Internet routers [1]. Next generation routers must be able to support thousands of optical links 32 each operating at 10 Gb/s (OC-192) or more [2]. Many techniques are available to perform IP address lookups[3 to 7]. Perhaps the most common approach in high-performance systems is to use Ternary Content Addressable Memory (TCAM) devices. While this approach can provide excellent performance, the performance comes at a fairly high price due to the exorbitant power consumption and high cost per bit of TCAM relative to commodity memory devices.

2 Proposed Approach

With using the prefix properties and Espresso-II algorithm for reduce the IP lookup table's rows Here, we propose an architectural technique that reduces the IP lookup table laterally. This technique adopts the multi-level routing lookup architecture applying the Multi Stage TCAMs, that we are called MSTCAM.

2.1 Routing Table Minimization

The figure1 depicted the schematic of minimization part of our architecture. Each EMU(Espresso Minimizer Unit) is as same as ROCM which is proposed in [8].

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 718–722, 2008.

2.1.1 Overlap Elimination

The overlap elimination technique eliminates redundant routing prefixes. The idea of overlap elimination is fairly simple. If *Pa* is an identical parent of *Pb*, then *Pb* is a redundant routing prefix.

2.1.2 Route Table Partitioning and PRTs

The partitioning partitions the prefixes based on their corresponding output port. For minimization we suppose that each subset as a separated table and call it Partial Route Table or PRT.

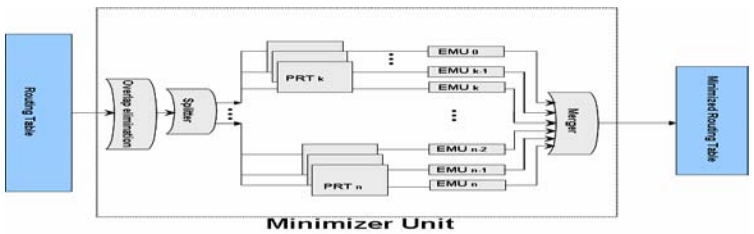


Fig. 1. Minimization unit (MU) schema

3 Proposed Architecture

In this architecture we propose using of MultiStage TCAM array instead of the general TCAM array structure. The proposed architecture is shown in the figure2.In a MSTCAM table, enabling of row cells is done stage by stage. Match Vectors are Match Line signals which are placed in the output of the last stage of each MSTCAM determine which row is matched. After doing the parallel lookup in MSTCAMs the final match vector is obtain by the dot product of segment match vectors. The LPM selector unit, dose the Longest Prefix Matching selection.

3.1 Lookup Operation

Figure3 shows the lookup operation by an activity diagram.Internal search mechanism in each MSTCAM is based on MLET approach and is described in [9] by us. In PLEA that depicted in the figure 2 worst case of time complexity is *8* cycle of CPU.

3.2 Update Operation

Update operation include two sub operations: insert and withdrawal. Our main objective in update operation is that the minimum part of TCAM table to be influenced. For any routing update, our approach restricts TCAM updates to a related PRT. So it is possible to update several PRTs simultaneously using multiple EMUs. Figure 4 and 5 depicted the activity diagram of insert and delete operations.

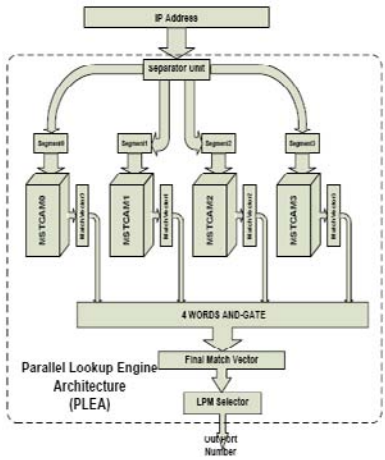


Fig. 2. Proposed architecture

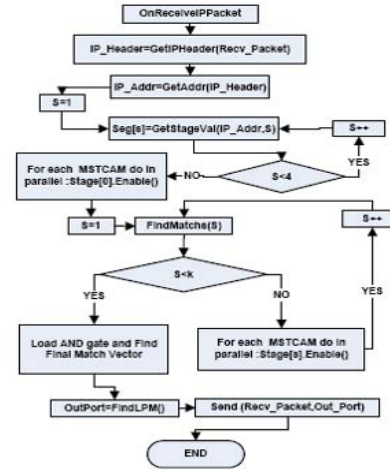


Fig. 3. Lookup activity diagram

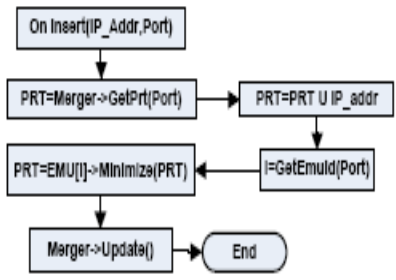


Fig. 4. Insert activity diagram

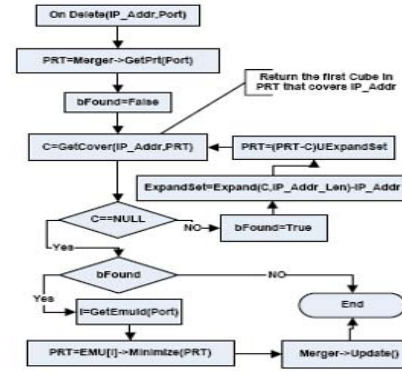


Fig. 5. Withdrawal activity diagram

4 Performance Evaluation

In this section we define some performance metrics and terms as below

EPS (Enabled bits Per Search): For example if the reference model [9] is used, we have: $EPS = S \times W$ (1)

Where S is the number of TCAM rows and W is the bit length of each row

MEPS (Mean Enabled bits Per Search):

$$MEPS = \sum_{i=1}^{m} EPS_i / m \quad (2) \quad \text{So } MEPS_{\max} = EPS_{\max} \quad (3)$$

POF (Power Optimization Factor): Suppose that a TCAM cell consume P watt of power when it is enabled so the POF of search for m address in TCAM table is:

$$POF = \frac{(MEPS_{\max} - MEPS) \times p}{MEPS_{\max} \times p} \times 100 = (1 - \frac{\sum_{i=1}^{m} EPS_i}{EPS_{\max}}) \times 100 = (1 - \frac{\sum_{i=1}^{m} EPS_i}{S \times W}) \times 100 \quad (4)$$

4.1 Experimental Results

We will describe the results in two aspects: the first is results of our minimization technique and the second is results of using PLEA.

4.1.1 Minimization Results

In our simulation 31000 prefixes of the existing prefixes in Telestra are given to the minimization unit as input set and after minimization process the prefix set decrease to the 12372 prefixes. Thus we could compact the TCAM table about 60 percentage.

4.1.2 PLEA Results

The following results (Figure6 and Table 1)obtain from testing the 10000 address from incoming address list of Telestra router which are selected randomly.

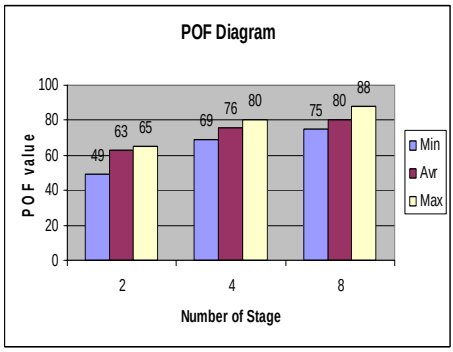


Fig. 6. POF of multiple configuration

Table 1. Performance comparison

Approaches		Minimization POF	Enabling POF	Total POF
PLEA	2-Stages	60	63	85.2
	4-Stages	60	76	90.4
	8-Stages	60	80	92
[2]		58	75	89.5
[6]		53	0	53
7]		0	36	36
[4]		0	28	28

References

1. Gupta, P.: Algorithms for Routing Lookups and Packet Classification. doctoral dissertation, Dept. Computer Science, Stanford Univ. (2000)
2. Ravikumar, V.C., Mahapatra, R.N.: TCAM Architecture for IP lookup Using Prefix Properties. IEEE Micro. 24(2), 60–69 (2004)
3. Taylor, D.E.: Models, Algorithms, and Architectures for scalable packet classification, Doctoral Thesis, Washington Univ. (August 2004)
4. Zane, F., Narlikar, G., Basu, A.: CoolCAMs: Power-Efficient TCAMs for Forwarding Engines. In: Proc. IEEE Infocom 2003, pp. 42–52. IEEE Press, Los Alamitos (2003)
5. Panigrahy, R., Sharma, S.: Reducing TCAM Power Consumption and Increasing Throughput. In: Proc. 10th Symp. High- Performance Interconnects (HOTI 2002), pp. 107–112. IEEE CS Press, Los Alamitos (2002)

6. Liu, H.: Routing Table Compaction in Ternary CAM. IEEE Micro. 22(1), 58–64 (2002)
7. Akhbarizadeh, M.J., Nourani, M.: An IP Packet Forwarding Technique Based on Partitioned Lookup Table. In: Proc. IEEE Int'lConf. Comm (ICC 2002), pp. 2263–2267 (2002)
8. Lysecky, R., Vahid, F.: On-chip logic minimization. In: Proceedings of the 40th conference on Design automation, pp. 334–337. ACM Press, New York (2003)
9. Mahini, A., Berangi, R., Mahini, H.: A Power Efficient Approach for Ternary Content Addressable Memory based, IP Lookup Engines in Internet Routers. In: CSICC 2007, pp. 748–755 (2007)

Assignment of OVSF Codes in Wideband CDMA

Mehdi Askari[1], Reza Saadat[2], and Mansour Nakhkash[2]

[1] Omeidyeh Islamic Azad University, Omeidyeh, Khuzestan, Iran
[2] Yazd University, Electrical Engineering Department, Yazd, Iran
Mehdiaskari58@yahoo.com

Abstract. In Wideband Code Division Multiple Access, Channelization codes are used to preserve the orthogonality between physical channels, in order to increasing system capacity. The Orthogonal Variable Spreading Factor (OVSF) codes are used as the channelization codes in this system. In WCDMA, it is possible to provide multi-rate service by employing the OVSF codes, which can be generated in the form of tree structure. This scheme is known as OVSF-CDMA. One important limitation of OVSF-CDMA is that the system must maintain the orthogonality among the assigned codes. The maintenance of the orthogonality among the assigned OVSF codes causes the code blocking problem. Efficient channelization code management, result in high efficiency of code utilization and increasing system capacity. This research compares the performance of OVSF code assignment schemes, in terms of code blocking probability and number of required code reassignment.

Keywords: WCDMA, Multi-rate service, Blocking Probability.

1 Introduction

In order to support variable rates of data multimedia in CDMA system, a set of orthogonal codes with different lengths must be used, because the rate of information varies and the available bandwidth is fixed [1], [2], [3]. It is possible to support higher data rates in direct sequence CDMA (DS-CDMA) systems by assigning multiple fixed-length orthogonal codes to a call. In an alternative CDMA scheme which is known as OVSF-CDMA, a single Orthogonal Variable Spreading Factor (OVSF) code is assigned to each user. In this case, a higher data rate can be accessed by using a lower spreading factor [4], [5]. The data rates provided are always a power of two with respect to the lowest-rate codes. OVSF codes assignment has significant impact on the code utilization of the system. The channelization operation in WCDMA transforms each data symbol into a number of chips. The number of chips per data symbol is called *spreading factor*. The data symbol are spread in channelization operation firstly and then scrambled in scrambling operation [6], [8].

There are two types of code assignment schemes: Static and Dynamic. This paper addressed both static and dynamic schemes in a WCDMA system where OVSF code tree are used. The general objective is to make the OVSF code tree as compact as possible so as to support more new calls by incurring less blocking probability and less reassignment costs.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 723–727, 2008.

2 OVSF Code System

The OVSF codes can be represented by a tree. Fig.1 shows a K layer code tree [4]. The OVSF code tree is a binary tree with K layer, where each node represents a channelization code (k,m), k=0,1,…,K, m=1,…,2^k. The lowest layer is the *leaf* layer and the highest layer is the *root* layer. The data rate that a code can be support is called its capacity. Let the capacity of the leaf codes (in layer K) be R. Then the capacity of the codes in layer $(K-1),(K-2),\ldots,0,1$ are $2R,4R,\ldots,2^{K-1}R,2^KR$ respectively, as shown in Fig. 1.

Layer K has 2^K codes and they sequentially labeled from left to right, starting from one. The m^{th} code in layer K is referred to as code (k,m). The total capacity of all the codes in each layer is 2^KR, irrespective of the layer number. We also define the maximum spreading factor $Nmax=2^K$ as the total number of codes in layer K. All lower layer codes spanned from a higher layer code are defined as *descendent* codes. All higher layer codes linking a particular code to the root code are called its *mother* codes [5].

Note that all codes in each layer are mutually orthogonal. Furthermore, any two codes of different layers are also orthogonal expect for the case when one of the two codes is the mother code of the other [2].

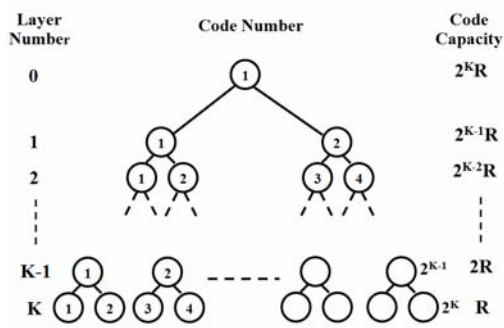


Fig. 1. A K-layer code tree

3 Problem Statement

When a new call arrives requesting for a code of rate iR, where i is a power of two, we have to allocate a free code of rate iR for it. In static schemes we address the allocation algorithm when multiple free codes exist in the code tree. When no such free code exist but the remaining capacity of the code tree is sufficient (i.e. summation of data rates of all free code is greater than iR), we can use dynamic schemes. In dynamic schemes, relocate some codes in the code tree to find a free space for the new call. OVSF code blocking define as the condition that a new call cannot be supported although the system has excess capacity to support the rate requirement of the call [5].

4 Static Code Assignment Schemes

In static schemes we address the allocation algorithm when multiple free codes exist in the code tree.

Ordered scheme: when a call requesting *iR* arrives, where *i* is a power of two, we need to find a code to accommodate this call. Our goal is to always vacate a larger capacity in the right-hand side of the code tree so as to accommodate higher-rate calls in the future [9]. So, if there is one or more than one free code in the code tree with a rate *iR*, pick the leftmost one in the code tree and assign it to the call. Otherwise, the call is rejected.

Least Assignability Branch scheme: the objective of this scheme is to keep the remaining assignable codes in the most compact state after each code assignment without reassigning codes. To achieve this purpose into the existing busy codes, new-code assignments are packed as tightly as possible into the existing busy codes. For a typical branch, say the branch under (k,m), let C(k,m) be the assignable capacity of the branch, which is defined as total capacity of the assignable leaf codes in this branch. In other word [4]:

$$C(k,m) = \sum_{i=1+(m-1).2^{K-k}}^{m.2^{K-k}} I_A^{(K,m)} \tag{1}$$

Where $I_A^{(K,m)}$ is the assignability index function of code (K,m) and is defined as:

$$I_A^{(K,m)} = \begin{cases} 1 & (K,m) \quad is \quad assignable \\ 0 & otherwise \end{cases} \tag{2}$$

When a call requesting *iR* arrives, where *i* is a power of two, we need to find a code to accommodate this call. If there is one or more than one code in the code tree with rate *iR*, pick the one whose ancestor codes has the least *C(k,m).* When there are ties, we will go one level up. If ancestors, has equal *C(k,m)*, we will follow the ordered scheme to assign the code.

5 Dynamic Code Assignment Schemes

Reassignment (Dynamic assignment) schemes are necessary when the capacity of the tree is enough to carry the incoming call, but no code of required rate is available.

Ordered scheme: The idea of this scheme is to rearrange some of the busy code and pack them as tightly as possible to one side of the tree. In doing so, the assignable codes are aggregated together. This method is simple, but it incurs many unnecessary code reassignments.

Optimal Scheme: Let a call requesting *iR* arrives to the system. In [5], a dynamic code assignment (DCA) algorithm is proposed based on *code pattern search* to find a branch of rate *iR* in the code tree which can be vacated with the minimum cost.

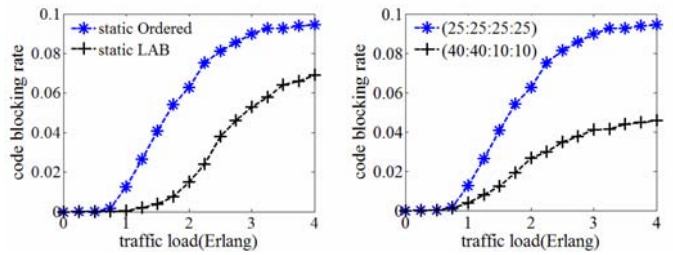
However, where to place those relocate codes is not addressed in [5]. In case a new call arrives requesting a rate *iR*, but no free code of such a rate exist, the following steps are taken:

1. If the total amount of free capacity in the code tree is greater than *iR*, apply the DCA algorithm in [5] to find the minimum-cost branch with rate *iR*. Otherwise, the call is rejected.

2. For the busy codes in the branch found above, we relocate them one-by-one starting from those with higher rates. For each busy code being relocated, we replace it based on LAB strategy. If no free codes exist to accommodate the transferred call, the same reassignment (DCA) procedure is repeated recursively until all busy codes are relocated.

6 Numerical Results

Our proposed schemes are evaluated on a 6-layer code tree. The call arrival process is modeled by a Poisson distribution, while the call duration is exponentially distributed. The possible rates for a new call are R, 2R, 4R, and 8R, each with a different probability of appearance. In our simulation we use a uniform rate distribution (R:2R:4R:8R=25:25:25:25) and a more realistic scenario with lower rate calls being more probable (R:2R:4R:8R=40:40:10:10) [5],[8],[9].To ensure stable results, each point on the figures has been produced by a simulation run with at least 10000



(a) (b)

Fig. 2. Code blocking rate due to traffic load (Erlang). (a) for different static schemes (b) for different code rate distribution at static Ordered scheme.

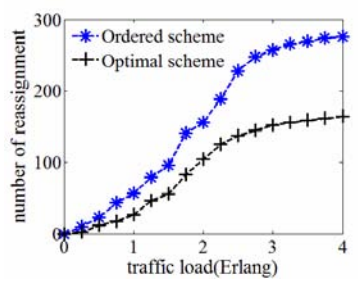


Fig. 3. Number of code reassignment in dynamic algorithms

incoming calls. Fig. 2(a) shows the code blocking probability at different traffic load for Static code assignment schemes and uniform rate distribution. We can see that LAB strategy performs the best and which is followed by the Ordered scheme. According to this Fig, at light load, the blocking probability is quiet insensitive to code placement algorithm. Fig.2(b) shows the code blocking rate for different code rate distribution at static Ordered scheme. According to this Fig, we can see that the blocking probability is the larger when the code rates are uniformly distributed. Fig. 3 shows the number of code reassignment for Dynamic algorithms. We can see that Ordered Dynamic algorithm has many code reassignments more than optimal algorithm.

7 Concluding Remarks

We have considered the problem of assignment single OVSF codes at the WCDMA third generation mobile communication systems. We have shown that OVSF codes assignment do have significant impacts on the code utilization of the system. The main idea is to keep the code tree less fragmented so as to accept more calls. So, we compare some static and dynamic code assignment schemes.

References

1. Adachi, F., Sawahashi, M., Suda, H.: Wideband DS-CDMA for next-generation mobile communication system. J. IEEE Communication Magazine 36, 56–69 (1998)
2. Dinan, E., Jabbari, B.: Spreading Codes for Direct Sequence CDMA and Wideband CDMA Cellular Networks. J. IEEE Communication Magazine 36, 48–54 (1998)
3. Adachi, F., Sawahashi, M., Okawa, K.: Tree structured generation of orthogonal spreading codes with different length for forward link of DS-CDMA mobile radio. Electronic Letters 33(1), 27–28 (1997)
4. Yang, Y., Yum, T.S.: Maximally Flexible Assignment of orthogonal variable spreading factor codes for multi-rate Traffic. Department of Information Engineering, Hong Kong (2001)
5. Minn, T., Siu, K.Y.: Dynamic assignment of orthogonal variable-spreading-factor codes in W-CDMA. J. IEEE Journal on Selected Areas in communications, 1429–1440 (2000)
6. http://www.3gpp.org
7. Lin, I.C., Gitlin, R.D.: Multi-Code CDMA Wireless Personal Communications Networks. In: Proc. ICC 1995, pp. 1060–1064 (1995)
8. Chen, J.C., Chen, W.: Implementation of an efficient channelization code assignment algorithm in 3G WCDMA. Institute of computer Science, Taiwan (2001)
9. Tseng, Y., Chao, C., Wu, S.: Code Placement and Replacement Strategies for Wideband CDMA OVSF Code Tree Management. J. IEEE transactions on mobile computing 1(4), 293–302 (2002)

Toward Descriptive Performance Model for OpenMosix Cluster

Bestoun S. Ahmed[1], Khairulmizam Samsudin[1], Abdul Rahman Ramli[1], and ShahNor Basri [2]

[1] Department of Computer & Communication Systems Engineering, University Putra Malaysia, 43400 Serdang, Selangor, Malaysia
bestoon82@gmail.com ,{kmbs,arr}@eng.upm.edu.my
[2] Department of Aerospace Engineering, University Putra Malaysia, 43400 Serdang, Selangor, Malaysia
shahnor@eng.upm.edu.my

Abstract. Computer cluster adoption has been slow because of the lack of single system image based cluster. OpenMosix is one instance of a successful SSI cluster. To date, there is no standard performance measurement metrics suitable for OpenMosix. A standard descriptive performance model benchmark allows developer and user to measure and compare performance of different SSI cluster design. The aim of this part of research is to study the effect of the number of nodes on some performance metrics by considering a particular network configuration.

Keywords: OpenMosix, MOSIX, Single System Image (SSI), Performance Modeling.

1 Introduction

The most important paradigm of SSI over other types of clusters are the needless of paralyze code, automatic process migration and the well aware of system resources [1].

There are many implementation of SSI, including OpenSSI and Kerrighed [2]. However, MOSIX [3, 4] and OpenMosix [5, 6] are promising for providing load-balancing operating system. However, the nonexistences of benchmark tools for SSI operating system cluster prevent performance measurement and comparison of the design. Standard benchmarks (e.g. for MPI and PVM) have different approach since they use a simple static job assignment algorithm that completely ignores the current state of dynamic load-balancing algorithm in OpenMosix [7].

2 OpenMosix Architecture

The preemptive process migration (PPM) is responsible for the resource management algorithm. As long as the requirements for CPU are below certain threshold point, all processes are confined to their Unique Home Node (UHN). When their

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 728–731, 2008.

requirements exceed the CPU threshold levels, the deputy [8] will migrate some processes transparently to other nodes [5, 4].

3 Cluster Construction

The OpenMosix cluster consists of eight COTS computer based on Intel Pentium 4, 1.6 GHz processor with 256 MB of main memory. There is no graphic card on the nodes except for the main node. The nodes are interconnected using an Ethernet 10/100 Mbits/s switch. GNU/Linux kernel 2.4.26 compiled with OpenMosix patch was installed on each node.

4 Results and Discussions

There are several parameters, which influence the performance modeling of SSI. In this paper, run-time, speedup and efficiency performance metrics will be considered [9]. Each performance metrics considered has been measured using DKG [10]. The program generates 4000 RSA public and private key pairs with 1024 bits.

Fig. 1 depicts the run-time of DKG with the increasing number of cluster nodes. We note that there is a significant performance improvement between one node and two nodes (nearly 50%); however, there is no significant improvement after having five nodes.

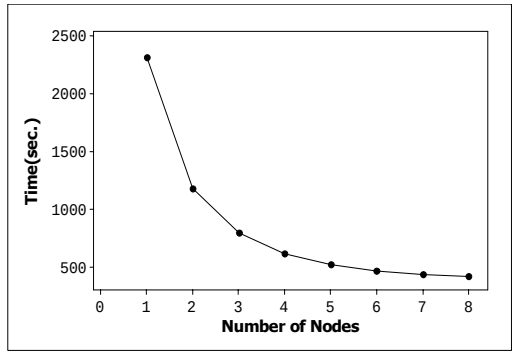


Fig. 1. Run Time of DKG on OpenMosix cluster

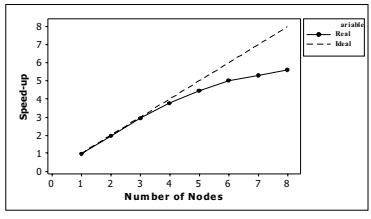


Fig. 2. Speed-up of DKG on OpenMosix

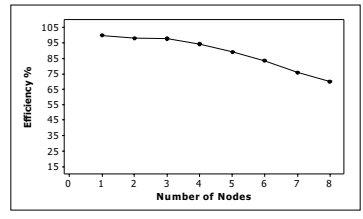


Fig. 3. Efficiency of DKG on OpenMosix

Fig. 2 and Fig. 3 shows the speed-up and efficiency of the cluster system. It is clear that the speed-up improvement is not linear with the number of nodes and the efficiency start to degrade after having more than three nodes.

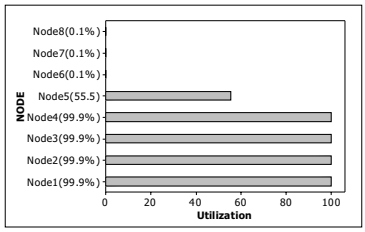


Fig. 4. CPU utilization with 4-child

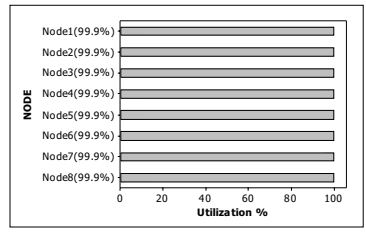


Fig. 5. CPU utilization with 8-child

To demonstrate the importance of having enough child process spawned from the parent, the peak CPU utilization is measured. Fig. 4 and Fig. 5 illustrates CPU utilization for each node in the cluster by executing DKG with 4-child and 8-child. Note that the load generated from 4-child DKG does not exceed OpenMosix CPU threshold and therefore the processes is not migrated to other nodes. Executing DKG with 8-child surpass the OpenMosix CPU threshold.

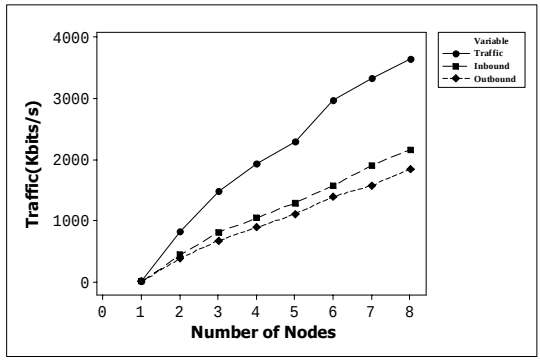


Fig. 6. Nodes Relation with Traffic

To evaluate the impact of network traffic on OpenMosix performance, the traffic on the home node executing 8-child DKG is measured using IPtraf [11]. Fig. 6. shows the actual inbound and outbound traffic of the home node. Adding additional node increases the traffic and as a result cause a noticeable drop of the cluster performance.

5 Conclusion

In this paper, partial description of the performance model has been achieved. Note that efficiency, speedup and even performance were inversely proportional to the number of nodes. The degradation of speed-down and efficiency is not caused by the

load threshold. The increasing network traffic can be explained partly by the Open-Mosix decentralized load-balancing algorithm [8]. This algorithms aim to improve scalability and fault tolerance. However, the load-balancing decision might not be the best as each node in the cluster manages its own information vector.

References

1. Christine, M., Pascal, G., Renaud, L., Geoffroy, V.: Toward an Efficient Single System Image Cluster Operating System. In: Future Generation Computer Systems, vol. 20(2). Elsevier Science, Amsterdam (2004)
2. Renaud, L., Pascal, G., Geoffroy, V., Christine, M.: Openmosix, OpenSSI and kerrighed: A Comparative Study. In: IEEE Int. Symp. on Cluster Comp. and the Grid (2005)
3. Barak, A., Láadan, O., Shiloh, A.: Scalable Cluster Computing with MOSIX for Linux. In: Proc. of Linux Expo 1999, Raleigh, N.C (1999)
4. Barak, A., Láadan, O.: The MOSIX Multicomputer Operating System for High Performance Cluster Computing. Journal of Future Generation Comp. Sys. 13(4-5) (1998)
5. Moshe, B., Maya, K., Krushna', B., Asmita, J., Snehal, M.: Introduction to OpenMosix, Linux Congress (2003)
6. OpenMosix website, `http://openmosix.org`
7. Keren, A., Barak, A.: Opportunity Cost Algorithms for Reducing of I/O and Interprocess Communication Overhead in a Computing Cluster. IEEE Trans. on Prallel and Distributed Systems 14(1) (2003)
8. Malik, K., Khan, O., et al.: Migratable Sockets in Cluster Computing. The Journal of Systems and Software 75, 171–177 (2005)
9. Rajkumar, B.: High Performance Cluster Computing, Australia, vol. I, pp. 70–71. P.H. PTR (1999)
10. Chen, Y.–H.: `http://ying.yingternet.com/mosix`
11. Iptraf, `http://iptraf.seul.org`

Analysis of the Growth Process of Neural Cells in Culture Environment Using Image Processing Techniques

Atefeh S. Mirsafian[1], Shirin N. Isfahani[1], Shohreh Kasaei[2], and Hamid Mobasheri[3]

[1] MSc Student of Computer Science, Sharif University of Technology
[2] Computer Engineering Department, Sharif University of Technology
[3] Institute of Biochemistry and Biophysics, University of Tehran

Abstract. Here we present an approach for processing neural cells images to analyze their growth process in culture environment. We have applied several image processing techniques for: 1- Environmental noise reduction, 2- Neural cells segmentation, 3- Neural cells classification based on their dendrites' growth conditions, and 4- neurons' features Extraction and measurement (e.g., like cell body area, number of dendrites, axon's length, and so on). Due to the large amount of noise in the images, we have used feed forward artificial neural networks to detect edges more precisely.

Keywords: Biophysics, cell culture, image processing, segmentation, noise reduction, classification, artificial neural networks, edge detection.

1 Introduction

The nervous system is one of the most critical and sensitive physiological system of human body which will hardly be regenerated by the body in case of damage. Thus, research in the field of curing spinal cord injuries is one of the most critical problems of medical science. Fortunately, in recent years some considerable attempts have been made which led to important results considering the injured. One step of the proposed treatment process is to culture neural cells in an external environment outside of the body under special conditions like electro-magnetic fields, chemical substances, special polymers, and so on. In this method, several images are taken from live neurons at some fixed intervals during culture process to study the effect of environment on their growth process. A large amount of such images shall be studied to obtain reliable statistics of the growth process of neural cells. Hence, performing this analysis manually is a very hard and time-consuming job. It may also have lots of computational errors. In this paper, we present our approach to analyze these images automatically using image processing techniques.

The images are taken from live neural cells in the culture environment. So, the quality of these images is very low. Three most effective reasons for this low quality are as follow: 1- Existence of environmental liquid in which the cells are cultured. 2- Food particles which are used to feed the cells. 3- Difference between dept of cells

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 732–736, 2008.

and dept of their branches from the microscope lens. Therefore, one of the most important steps of our work is to reduce images' noise with minimum loss of details. Segmenting cells, extracting cell bodies and measuring number and length of dendrites are the next steps.

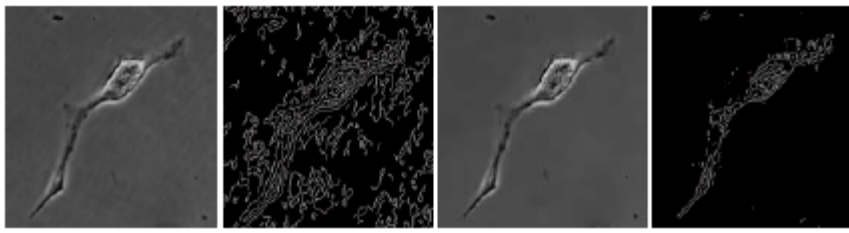
2 Noise Reduction Method

The effect of environmental liquid and limitations of light microscope imaging make live cells' images unclear. Thus, it is so hard to detect cells' membrane precisely. Employing an appropriate method of noise reduction, therefore, has a large effect on the quality of analysis results. Figures 1.a and 1.b illustrate one of studied images together with the output of Canny edge detector mask on it. As it is clear from these images the effect environmental liquid originates a lot of false detection.

Furthermore, neurons' axons are not always clear enough to be distinguished from noise. Thus, we need an algorithm that keeps the details as much as possible while reducing the noise. Different algorithms were studied for this purpose. Among them, we can point to spatial filters [1], frequency domain filters [1], non-local means algorithm [2], noise reduction using wavelet packet [3], and noise reduction using curvelet [4]. None of these methods could successfully meet our goals. The applying method [Peters 5] is an adaptive noise reduction algorithm using mathematical morphology. This method is a new Morphological Image Cleaning algorithm which generalizes OCCO smoothing function defined as follows:

$$OCCO\ (f,z) = \frac{1}{2}\left[(f_z)^z + (f^z)_z\right]. \tag{1}$$

where f_z represents the morphological opening of the image, f, with Structural Element (SE) Z and f^z represents the morphological closing of the image with Z. The applying method smoothes the images with several OCCO filters of different diameters. Let S_j be the result of smoothing the image with an OCCO filter using SE of diameter d_j. Let $S_0 = f$. Assume $d_0 < d_1 < \ldots < d_k$ and that $d_0 = 1$. Then, $D_j = S_{j-1} - S_j$ is the j'th residual image. D_j has, for the most part, no features larger than d_j and none smaller than d_{j-1}. It is assumed to contain both features and noises. By determining appropriate



(a) (b) (c) (d)

Fig. 1. a) A sample neuron image. b) Canny filter output on 1.a. c) Result of noise reduction. d) Canny filter output after noise reduction.

threshold for each D_j (using its statistical specifications), we added back some features to S_k. Figures 1.c and 1.d illustrate the output of this algorithm using image of Fig. 1.a as input together with the output of Canny edge detector mask on it. The remarkable change in edge detector result proves that the method meets our goals.

3 Image Segmentation

It is necessary to consider the following issues in segmentation of neuron's image:

- A bright shadow usually exists around the cell body because of its shape which makes it acts like a lens. This shadow is not a part of neuron and shall be removed.
- Cell's wall is light in some parts of neurons and dark in other parts. Considering the bright shadow, it causes ambiguity in edge detection process.

Here, we applied a hybrid segmentation method to increase the accuracy and to reduce the effect of issues which was mentioned. This method consists of a combination of Graph Segmentation and Segmentation using Neural Networks.

3.1 Graph Segmentation

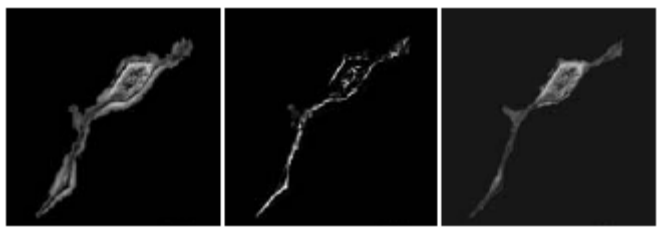
In this method, edge elements are considered as a graph. At the beginning, each point of (x, y) in image is considered as a graph node and the points of (x, y+1), (x+1, y), (x+1, y+1) and (x-1, y+1) are its adjacent nodes in the graph. The weight of the edge between every two points of p and q is calculated as follows:

$$\sqrt{(red\,(p) - red\,(q))^2 + (green\,(p) - green\,(q))^2 + (blue\,(p) - blue\,(q))^2}\,. \tag{2}$$

The segmentation algorithm merges components weight of edges between of which has small value compared with internal edges of those components. The output of this method does not segment the cells completely and cannot be used just by itself (Fig. 2.a). However, it can segment an approximate region used in next step.

3.2 Segmentation Using Artificial Neural Networks

Because of the special conditions of edges in live cells' images, we applied Artificial Neural Networks to detect edges, which helped us increase accuracy. In this method,



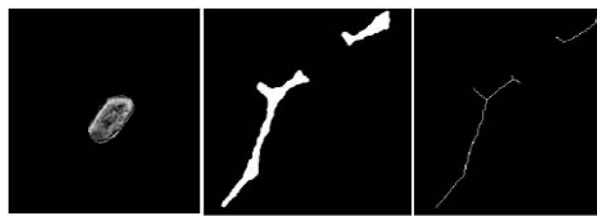
(a) (b) (c)

Fig. 2. a) Graph segmentation output. b) Neural Network output. c) Final segmentation result.

for every point of (*x, y)* in the image, a 7×7 window around it is considered as Network input. The output is also the probability of the point of (*x, y*) being edge. It is a value between 0 and 1. After training, the network calculates the probability of being edge for each point. We used the output of Graph Segmentation method as Network's input. An example of Network's output is shown in Fig. 2.b. In this image, each point has a value between 0 and 1. So, an appropriate threshold, calculated based on output's statistical characteristics, will specify edges. An example of final segmentation output is shown in Fig. 2.c.

4 Cells' Feature Measurement

Measuring of some cells' features such as cell body area, number of dendrites, and length of dendrites are critical for analysis of cells' growth process. Here, we used morphological opening operator to remove dendrites and to simplify measuring cell body area (Fig. 3.a). We also thinned dendrites using morphological thinning operator to measure their length (Fig. 3.b and Fig. 3.c).



(a) (b) (c)

Fig. 3. a) Cell body. b) Dendrites. c) thinned dendrites.

5 Experimental Results

Even though some few research have been reported [6, 7], as we searched and studied, there is no other similar work to be compared with presented approach. Thus, we examined the applied segmentation method to measure its efficiency. We evaluated different segmentation methods using:

$$\sigma_1 = \frac{\text{number of wrong segmented points}}{\text{sum of exact segments area}} \tag{3}$$

$$\sigma_2 = \frac{\text{number of correct segmented points}}{\text{sum of exact segments area}} \tag{4}$$

$$\sigma_3 = 1 - \sigma_2 + \sigma_1 . \tag{5}$$

"Wrong segmented points" in (3) means the points which were not exactly part of cells but were segmented incorrectly and σ_3 o represents the total error for each method. As it can be seen in Table 1, the proposed method resulted in minimum error.

Table 1. Performance Comparison of different segmentation methods

Method	σ1	σ2	σ3
Graph segmentation method	0.8073	0.9917	0.8156
Segmentation using Canny edge detector	0.4627	0.9567	0.5060
Segmentation using Sobel edge detector	0.2721	0.8892	0.3829
Proposed segmentation method	0.1395	0.9709	0.1686

6 Conclusion

In this paper, we proposed an efficient approach for analyzing images of neural cells using image processing techniques. Segmentation and analysis of microscopic images is very challenging because of their and low quality and large amount of noise. This process becomes more difficult for live neural cell images which are taken in the culture environment. To decrease the effect of ambiguity while segmenting neurons, we proposed a hybrid segmentation method, which gave promising results.

References

[1] Gonzalez, R.C., Woods, R.E.: Digital Image Processing, 2nd edn. Prentice Hall, Englewood Cliffs (2002)
[2] Buades, A., Morel, J.M.: A non-local algorithm for image denoising. In: International Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 60–65 (June 2005)
[3] Wu, Z., Shengsheng, Y., Jingli, Z.: A Method of Image Noise Reduction Based on Multi-Wavelet Transform and Multiscale Data Fusion. Transaction on Engineering, Computing and Technology 2 (December 2004) ISSN 1305-5313, ENFORMATIKA
[4] Eslami, R., Radha, H.: Image Denoising using translation-invariant contourlet transform. In: IEEE International Conference on Acoustics Speech and digital processing, vol. 4, pp. 557–560 (March 2005)
[5] Peters, R.A.: A New Algorithm for Image Noise 'reduction using Mathematical Morphology. IEEE Transactions on Image Processing 4(3) (May 1995)
[6] Xu, X., Chen, X., Liu, K., Zhu, J., Zhou, X., Wong, S.: A Computer-based System to Analyze Neuron Images. In: IEEE International Symposium on Circuits and systems, vol. 5, pp. 4225–4228 (May 2005)
[7] Young, D., Gray, A.J.: Cell identification in differential interference contrast microscopic images using edge detection. In: Proc. 7th British Machine Vision Conference, Edinburgh, vol. 1, pp. 133–142. BMVA Press (1996)

Bandwidth-Delay Constrained Least Cost Multicast Routing for Multimedia Communication

Mehrdad Mahdavi[1], Rana Forsati[2], and Ali Movaghar[1]

[1] Department of Computer Engineering, Sharif University of Technology, Tehran, Iran
[2] Department of computer Engineering, Islamic Azad University, Karaj Branch, Karaj, Iran
mahdavi@ce.sharif.edu, forsati@kiau.ac.ir, movaghar@sharif.edu

Abstract. A new meta-heuristic algorithm is proposed for constructing multicast trees in real-time multimedia applications such that satisfies two important Quality of Service (QoS) constraints. The QoS based multicast routing problem is a known NP-complete problem that depends on (1) bounded end-to-end delay and link bandwidth along the paths from the source to each destination, and (2) minimum cost of the multicast tree. We evaluate the performance and efficiency of the proposed method with a modified version of the BSMA algorithm which is the best known deterministic heuristic algorithm to delay-constrained multicast problem. Simulation results reveal that the proposed algorithm can achieve a smaller average tree costs than modified BSMA with a much smaller running time for relatively large networks.

Keywords: Multicast, QoS routing, multimedia, harmony search.

1 Introduction

Multicasting involves the transport of the same information from a sender to multiple receivers along a multicast tree with varying quality of service (QoS) constraints. Generally, there are two approaches to solve this problem: (1) an optimal solution in exponent time; (2) a near-optimal solution by a heuristic algorithm. Though (1) is an optimal solution, it is impractical due to its NP hard computation and requires large amount of computation time. Instead (2) is a feasible way and some heuristics such as simulated annealing (SA) [1], genetic algorithms (GA) [2] for the delay-constrained or more constraints multicast routing problem have been studied in the literatures.

The problem formally can be defined as follows. A network is modeled as a directed weighted graph $G=(V,E)$ where V is a finite set of nodes and E is a set of edges representing communication links between network nodes. Three non-negative real-valued functions are associated with each link $e(e\in E)$: cost $C(e):E\to R^+$, delay $D(e):E\to R^+$ and available bandwidth $B(e):E\to R^+$. $C(e)$ defines the measure we want to optimize, $D(e)$ and $B(e)$ define the criteria we want to constrain (bound).

In this paper, we apply harmony search [3] algorithm to solve bandwidth-delay-constrained least-cost multicast routing. The proposed algorithm is compared with the

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 737–740, 2008.

deterministic delay-constrained multicast routing, namely the Bounded Shortest Multicast Algorithm (BSMA) [4] considering trees cost and execution time.

The paper is organized as follows: Section 2 describes the proposed HS-based algorithm. The simulation results are presented in section 3. Section 4 concludes the paper.

2 The Proposed Algorithm

2.1 Preprocessing

Since, each edge that violates the bandwidth constraint can not be present in any path from source node to each destination node and can be eliminated, so, by eliminating such edges, the problem is converted to delay-constrained least cost one and ensures that we will not have a bandwidth-violated path in the graph.

2.2 Encoding

The encoding in our proposed algorithm is similar to that used in [6] for solving Steiner problem in a graph. Each solution vector in HM specifies a set of selected Steiner nodes. The selected Steiner nodes are specified by a bit-string in which each bit corresponds to a specific node. Assume a fixed numbering 0, 1, …, (n-m-2) of the Steiner nodes V-S-{s}, a solution vector is the set : { g(0), g(1), …, g(n-m-2)} where $g(i) \in \{0,1\}, i = 0,1,...,(n-m-2)$. The set of Steiner nodes $S_T \subseteq V - S - \{s\}$ specified by the solution vector is $S_T = \{v_i \in V \mid g(i) = 1, i = 0,1,...,n-m-2\}$.

2.3 Decoding

According to our representation, each solution vector gives a set of Steiner nodes that can be present in multicast tree. Let S_T represents these nodes. The proposed decoder is based on a deterministic delay-constrained multicast routing algorithm (KMB) proposed in [7]. Each solution vector is decoded as follow. First, construct a graph G' with the nodes being $\{s\} \cup S \cup S_T$. Each link (v_i, v_j) in G' is a cheapest path from v_i to v_j in original network $G = (V, E)$ satisfying the delay-constraint. Then construct a delay-constrained spanning tree T' of G'. The tree T' rooted at s and spanning all nodes in $S \cup S_T$. In the next step each edge in T' is converted to the corresponding paths they represent. Finally, leaf nodes in T' that are not in D must be pruned. Due to KMB a delay-constrained multicast tree T rooted at s, the decoder also do. The computational complexity of the decoder is therefore $O((m + |S_T|)n^2)$.

2.4 Evaluation of Solutions

The delay constrained is considered here. Let $F(T(s,D))$ is a function that return fitness of multicast tree $T(s,D)$ and has the property that, if the $T(s,D)$ is

delay-constrained tree, then, it must return cost of tree as it fitness value. The most common approach to handle constraints is to use penalties in $F(T(s,D))$. According to these properties, we use following relation to evaluate solutions:

$$F(T(s,D)) = \frac{\alpha}{\sum_{e \in T(s,D)} C(e)} \prod_{d \in D} \Phi(Delay(P(s,d) - \Delta_d), \quad \Phi(z) = \begin{cases} r & z > 0 \\ 1 & z \le 0 \end{cases} \tag{5}$$

where α is a positive real factor, $\Phi(z)$ is a penalty function as follow: when the solution satisfies delay constraint, the value of $\Phi(z)$ is 1 and otherwise is r. We select r = 0.5 in our experiments.

2.5 Harmony Operations

Each solution vector based on over representation is an array of length *(n-m-1)* of bit strings. For initialization of **HM**, each row is filled with randomly generated solution vectors. We modified *PAR* parameter as follows. In this computation, *PAR* became the rate of moving to the nearest adjacent node from one node. We define three *PAR's* ($PAR_1 = 0.6 \times PAR$, $PAR_2 = 0.3 \times PAR$, *and* $PAR_3 = 0.1 \times PAR$) that are the rates of moving to constrained nearest, second nearest and random adjacent node respectively. Let *v* is node that we want apply PAR and u is currently adjacent node to *v*. The node *u* is replaced with another node *w* if the delay of link *(v, w)* is less than or equal to delay of link *(v, u)* and the cost of *(v, w)* is less than *(v, u).*

3 Performance Analysis

To generate random networks, we have used a random graph generator based on the Salama graph generator [8]. We have performed extensive simulation studies on various randomly generated networks with 10–100 nodes to evaluate the performance of the proposed HS-based algorithms. In all of the randomly generated networks, the size of the multicast group is considered as 20% of the number of network nodes. The multicast group is randomly selected in the graph. For all of the simulations HMCR = 0.9, PAR_{min}=0.01, and PAR_{max} = 0.95. In the first set of experiments, our proposed HS-based algorithm is compared with the BSMA for cost performance. A preprocessing step has been added to BSMA for removing edges that are violating bandwidth constraint. Fig. 1 shows the cost performance for different numbers of network nodes (from 10 to 100 in steps of 10, group size =20% of number of nodes in network). It can be seen from Fig. 1 that our proposed algorithm has result in a smaller average tree cost than the BSMA. Also as the network size increases the difference between two algorithms increases which means that our proposed algorithm performs better with the increasing of network size.

In the second set of experiments, our proposed algorithm is compared with the mentioned modified BSMA for the execution time. Fig. 2 shows the average execution time of algorithms. In Fig. 2, it can be seen that our proposed algorithm has a much better running time performance than BSMA for relative large networks.

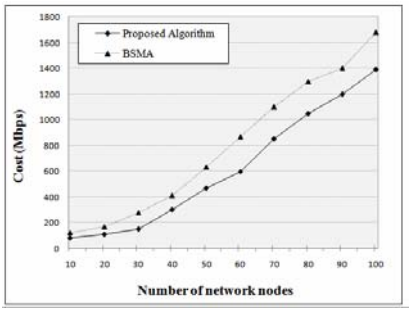


Fig. 1. Tree cost versus network size, group size = 20 %, and Δ = 30 ms

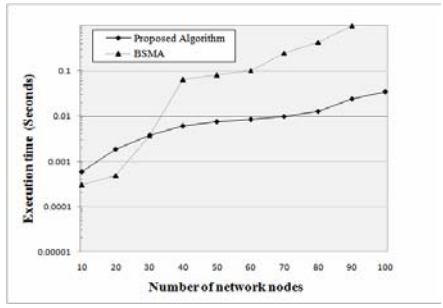


Fig. 2. Execution time of algorithms for different network sizes, multicast group size is 20% of number of nodes

4 Conclusion

We studied the problem of constructing bandwidth-delay constrained least cost multicast trees and presented an algorithms based on HS for obtaining such trees. The proposed algorithm is compared against near-optimal heuristic BSMA algorithm considering tree cost and execution time and achieved a smaller tree costs with a much smaller execution time. In the proposed algorithm each solution vector specifies a set of Steiner nodes which are represented by a bit-string.

References

1. Kun, Z., Yong, Q., Hong, Z.: Dynamic Multicast Routing Algorithm for Delay and Delay Variation-Bounded Steiner Tree Problem. Knowledge-Based Systems 19(2006), 554–564 (2006)
2. Wang, X., Cao, J., Cheng, H., Huang, M.: QoS Multicast Routing for Multimedia Group Communications Using Intelligent Computational Methods. Computer Communications 29(12), 2217–2229 (2006)
3. Lee, K.S., Geem, Z.W.: A New Meta-Heuristic Algorithm for Continues Engineering Optimization: Harmony SearchTheory and Practice. Comput. Methods Appl. Mech. Engrg
4. Parsa, M., Zhu, Q., Garcia-Luna-Aceves, J.J.: An Iterative Algorithm for Delay-Constrained Minimum-Cost Mmulticasting. IEEE/ACM Transactions on Networking 6(4) (1998)
5. Wang, Z., Crowcroft, J.: Quality of Service for Supporting Multimedia Applications. IEEE Journal on Selected Areas in Communications 14(7), 1228–1234 (1996)
6. Esbensen, H.: Computing Near-Optimal Solutions to the Steiner Problem in a Graph Using Genetic Algorithm. Networks 26, 173–185 (1995)
7. Kou, L., Markowsky, G., Berman, L.: A Fast Algorithm for Steiner Trees. Acta Informatica 15 (1981)

Cellular Probabilistic Evolutionary Algorithms for Real-Coded Function Optimization

M.R. Akbarzadeh T. and M. Tayarani N.

Azad University of Mashhad, Iran
Akbarzadeh@ieee.org, tayarani@ieee.org

Abstract. We propose a novel Cellular Probabilistic Evolutionary Algorithm (CPEA) based on a probabilistic representation of solutions for real coded problems. In place of binary integers, the basic unit of information here is a probability density function. This probabilistic coding allows superposition of states for a more efficient algorithm. Furthermore, the cellular structure of the proposed algorithm aims to provide an appropriate tradeoff between exploitation and exploration. Experimental results show that the performance of CPEA in several numerical benchmark problems is improved when compared with other evolutionary algorithms like Particle Swarm Optimization (PSO) and Genetic Algorithms (GA).

Keywords: Evolutionary Algorithms, Probabilistic Evolutionary Algorithms, Optimization.

1 Introduction

Quantum Evolutionary Algorithm (QEA) is an approach in which chromosomes are coded after quantum states of electrons in a probabilistic fashion. In this architecture each chromosome consists of m Q-bits that are equivalent to a superposition of 2^m states. This characteristic makes the diversity of the population high and improves the exploration of the algorithm without loss of exploitation. In [4, 5] quantum-inspired evolutionary algorithms are investigated for a class of combinatorial optimization problems in which quantum rotation gates act as update operators. This quantum rotation gate is also used in a novel parallel quantum GA for hierarchical ring model and infinite impulse response (IIR) digital filter design [6]. Reference [7] proposes a quantum evolutionary algorithm for multi-objective optimization and quantum rotation gate.

While the above algorithms have reported good success in application of QEA and its variants on several problems, quantum representation is only applicable to problems with binary coding. For real coded problems, the problem in real domain must first be mapped to a binary coding before optimization by QEA. This approximation can introduce undesirable limitations and errors for QEA on real coded problems.

Here, we propose a novel evolutionary algorithm for real coded problems that, similar to QEA, has a probabilistic structure that aims to take advantage of superposition of states. Furthermore, a cellular interaction architecture is proposed that

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 741–744, 2008.

promotes local interaction / leadership for better exploitation/exploration of local neighborhoods.

This remainder of this paper is organized as follows. In Section 2 we propose our novel algorithm called PEA. The experimental results are discussed in Section III.

Table 1. Lookup table of $\Delta\mu$. *f(x)* is the fitness of real solution x^t_{ij} and *f(b)* is the fitness of B_{ij}

$x_{ij} > b_{ij}$	$f(x) \geq f(b)$	$\Delta\mu$
False	*False*	0.01
False	*True*	-0.01
True	*False*	-0.01
True	*True*	0.01

2 The Proposed Cellular Probabilistic Evolutionary Algorithm

The proposed Probabilistic EA uses a new representation for solutions called P-value. A P-value individual is a string of P-values. A P-value is defined as $[\mu,\delta]^T$. Where μ is the mean and δ is the width of a uniform probability density function. The value of μ is in the range of $l \leq \mu \leq u$, where l and u are the lower and upper bounds of the search space respectively. A P-value individual is defined as a string of P-values. This probabilistic representation makes the diversity of the population higher than in other non-probabilistic representations.

This paper also uses a cellular structure for PEA. The structure of the CPEA is shown in Fig. 1. The advantage of cellular structure is that the fitness and genotype diversity in the population is preserved for a long number of generations [1]. The pseudo-code of CPEA is considered as below.

```
Procedure CPEA
begin
  t=0
1.initialize P(0).
2.make X(0) by observing the values of P(0).
3.evaluate X(0).
4.for all real solutions x_ij^0 in X(0) do
   begin
5.  find N_ij in X(0).
6.  select solution x with best fitness in N_ij and store it in B_ij
   end
7.while not termination condition do
   begin
    t=t+1
8.  observe P(t-1) and make X(t)
9.  evaluate X(t)
10. update P(t) based on B_ij and X(t)
          using update operator.
11. for all real solutions x_ij^t in X(t) do
     begin
12.   find N_ij in X(t).
13.   select binary solution x with
            best fitness in N_ij.
14.   if x is fitter than B_ij save x in
            B_ij.
```

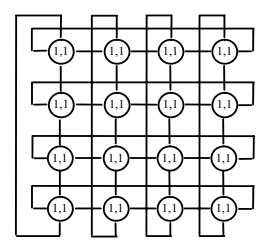


Fig. 1. The cellular structure of CPEA

```
15.     if R(0,1)<r reinitialize p_ij^t randomly.
        end
    end
end
```

The population of probabilistic individuals is represented as $P(t) = \left\{ p_{ij}^t \mid i, j = 1,2,...,S \right\}$, where t is generation number and S is the size of lattice-like population. The probabilistic individual p_{ij}^t is defined as:

$$p_{ij}^t = \begin{bmatrix} \mu_{ij,1}^t & \mu_{ij,2}^t & \cdots & \mu_{ij,k}^t & \cdots & \mu_{ij,m}^t \\ \delta_{ij,1}^t & \delta_{ij,2}^t & \cdots & \delta_{ij,k}^t & \cdots & \delta_{ij,m}^t \end{bmatrix} \tag{1}$$

Where m the string length of the P-valued individuals, and t is the generation number. The procedure of CPEA is described as below:

1. In the initialization step we set: $[\mu^0_{ij,k} \quad \delta^0_{ij,k}]^T = [\mathrm{R}(u_k, l_k) \quad u_k - l_k]$, for $i,j=1,2,\ldots,S$ and $k=1,2,\ldots,m$. where S is the size of lattice-like population, l_k, u_k are the lower bound and the upper bound of k-th dimension of search space and.
2. In the observe step, the real valued solutions $X(0)=\{x^0_{ij} | i,j=1,2,\ldots,S\}$ are evaluated by observing the probabilistic individuals $P(0)=\{p^0_{ij} | i,j=1,2,\ldots,S\}$. The R(.,.) is a uniform random number generator observing operation is performed as $x_{ij,k}^{t+1} = \mathrm{R}(\mu_{ij,k}^t - \delta_{ij,k}^t, \mu_{ij,k}^t + \delta_{ij,k}^t)$. Here each real valued individual x_{ij}^t is a string of real values which is a possible solution for the real coded problem.
3. In this step all the real valued individual's x_{ij}^0 are evaluated with fitness function.
4. In this step the "for" loop is run for all real solutions x_{ij}^0.
5. The neighbors of real solution x_{ij}^0 are found and stored in N_{ij}
6. The best real solution x in N_{ij} is found and stored to B_{ij}.
7. The algorithm is running until termination condition is satisfied.
8. The probabilistic population of individuals $P(t\text{-}1)$ is observed.
9. The real solutions are evaluated by fitness function.
10. In this step $P(t)$ is updated based on the values of $B(t)$ and $X(t)$. The update operator is performed as $[\mu_{ij,k}^{t+1} \quad \delta_{ij,k}^{t+1}]^T = [\Delta\mu \times (u_k - l_k) \times \mathrm{R}(0,1) \quad \delta_{ij,k}^t \times \eta]^T$ for $i,j=1,2,\ldots,S$ and $k=1,2,\ldots,m$. Where $0<\eta<1$ and $\Delta\mu$ is considered as Table. I.
11. In this step the "for" loop is running for all real solutions x_{ij}^t.
12. The neighbors of real solutions x_{ij}^t are found and stored in N_{ij}.
13. The best real solution x in N_{ij} is found.
14. If x is fitter than B_{ij} store it to B_{ij}.
15. In CPEA we consider the reinitialization operator to explore the search space. In this step if reinitialization condition is satisfied, the probabilistic individual p_{ij}^t is reinitialized.

3 Experimental Results

The population size for all of the experiments is set to 25. All results are averaged over 50 runs. The value of η is set to 0.99 and the reinitialization probability is set to 0.001. In conventional GA the value of 0.1 is used for mutation rate and 0.7 for two-point crossover. The PSO algorithm is considered as in [9], c_1, c_2=0.001×(u_i - l_i) and w=0.8. Here 14 benchmark numerical function optimization is used to evaluate the proposed algorithm as shown in Table II, for m=100.

Table 2. Experimental results of the numerical function optimization problems. MEAN and STD represent the mean best standard deviation of 50 runs respectively.

	CPEA		GA		PSO			CPEA		GA		PSO	
	Mean	Std	Mean	Std	Mean	Std		Mean	Std	Mean	Std	Mean	Std
f_1 [2]	2.7×10^{4}	1.5×10^{3}	3.6×10^{4}	5.1×10^{2}	7.4×10^{3}	1.4×10^{3}	f_8 [3]	188.5	1.5	184.1	1.2	144.7	2.8
f_2 [2]	-142.4	16.6	-196.8	19.7	-919.1	38.1	f_9 [2]	134.1	7.6	-4.1×10^{3}	736.4	-3.1×10^{4}	4.7×10^{3}
f_3 [2]	-0.6	0.06	-6.2	0.53	-17.4	0.62	f_{10} [2]	-0.01	0.006	-0.44	0.05	-1.39	0.08
f_4 [2]	0.33	0.036	-0.42	0.089	-2.9	0.403	f_{11} [2]	-6.27	0.75	-78.7	3.2	-53.3	7.7
f_5 [2]	57.4	5.2	-7.1×10^{3}	1.3×10^{4}	-6.3×10^{6}	3.7×10^{6}	f_{12} [10]	-3.2	6.6	-4.8×10^{3}	2.6×10^{3}	-1.27×10^{5}	3.7×10^{4}
f_6 [2]	-25.1	8	-7.6×10^{5}	1.0×10^{6}	-4.2×10^{7}	1.9×10^{7}	f_{13} [3]	-1.0×10^{2}	0.6	-7.6×10^{2}	1.2×10^{2}	-1.6×10^{3}	2.3×10^{2}
f_7 [10]	15.7	1.6	63.3	2.8	11.6	1.2	f_{14} [3]	-0.02	0.005	-0.12	0.07	-22.6	1.69

References

1. Alba, E., Dorronsoro, B.: The Exploration/ Exploitation Tradeoff in Dynamic Cellular Genetic Algorithms. IEEE Trans. Evol. Comput. 9, 126–142 (2005)
2. Zhong, W., Liu, J., Xue, M., Jiao, L.: A Multi-agent Genetic Algorithm for Global Numerical Optimization. IEEE Trans. Sys, Man and Cyber. 34, 1128–1141 (2004)
3. Koumousis, V.K., Katsaras, C.P.: A Saw-Tooth Genetic Algorithm Combining the Effects of Variable Population Size and Reinitialization to Enhance Performance. IEEE Trans. Evol. Comput. 10, 19–28 (2006)
4. Han, K., Kim, J.: Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. IEEE Trans. on Evolutionary Computation 6 (2002)
5. Han, K.H., Park, K.H., Lee, C.H., Kim, J.H.: Parallel quantum-inspired genetic algorithm for combinatorial optimization problem. In: Proceeding of the 2001 IEEE congress on Evolutionary Computation Seoul, Korea (2001)
6. Zhung, G., Jin, W., Hu, L.: Novel parallel quantum GAs. In: Proceedings of the 4th International Conference on Parallel and Distributed Computing, Applications and Technologies, PDCAT (2003)
7. Zhung, G., Jin, W., Hu, L.: Quantum evolutionary algorithm for multi-objective optimization. In: Proceedings of the 2003 IEEE International Symposium on Intelligent Control Houston, Texas, October 5-8 (2003)
8. Han, K.H., Kim, J.H.: Quantum-Inspired Evolutionary Algorithms with a New Termination Criterion, -Gate, and Two-Phase Scheme. IEEE Trans. Evol. Comput. 8, 156–169 (2004)
9. Alrashidi, M.R., EI-Hawary, M.E.: A Survey of Particle Swarm Optimization Applications in Electric Power Systems. IEEE Trans. Evol. Comput. 9, 1–6 (2005)
10. Khorsand, A.-R., Akbarzadeh-T, M.-R.: Quantum Gate Optimization in a Meta-Level Genetic Quantum Algorithm. In: IEEE International Conference on Systems, Man and Cybernetics (2005)

A New Approach for Scoring Relevant Documents by Applying a Farsi Stemming Method in Persian Web Search Engines

Hamed Shahbazi, Alireza Mokhtaripour, Mohammad Dalvi, and Behrouz Tork Ladani

Department of computer engineering University of Isfahan, Isfahan, Iran
{Shahbazi,Mokhtari,Dalvi,Ladani}@eng.ui.ac.ir

Abstract. In this paper, we will introduce a new approach for scoring Farsi (also called Persian) documents in a Persian Search engine. This approach is based on a new stemming method for Farsi language. Our new stemming method works without any dictionary. Evaluation results show significant improvement in performance (precision/ recall) of the Information Retrieval (IR) system using this stemmer. we have combine our stemming method with a mathematical scoring approach named FDS to obtain a powerful scoring policy for relevant documents in a Persian search engine.

Keywords: Information retrieval, Fourier, scoring, stemming, Persian, search engine.

1 Introduction

With informatics revolution the great deal of information produce in the entire world in every second.

For this reason one of the important challenges in computer science is information managing that need to it will appear while encountering to text data's.

Farsi is an Indo-European language that is spoken and written primarily in Iran, Tajikistan and parts of Afghanistan. Form of penmanship of Farsi contains 32 letters and is written from right to left. Some other languages like Arabic, Kurdish, and Urdu use this form of penmanship but have their own specifications. Farsi also has its own specifications such as not using accents (except in special cases) and polymorphism in writing. In this article we introduce an information retrieval (IR) model that is optimized for Persian data search.

1.1 Fourier Domain Scoring (FDS) Logic

Assume that the following question is asked to system:

"Does computer engineering have any effect on society?"

Naturally, union of all the good and keywords set in this question can be answers. Really this is a Boolean model ,but in his way ,incoherent (no-relative) texts also will returned .For example ,a text that is relative to narcotic materials effects on human

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 745–748, 2008.

societies and relative statistics of this phenomenon that collected in computer engineering union of university also are appears inside the answer. Although for lack of the similarity value there is no classification what so ever. If this example to be generalized, we understand that scattering keywords of a text has a special effect on produce its concept. About above case, evidently, in almost relative text, four keywords are mentioned near together.

Now, assume text that includes some paragraphs. Heavy text usually because their meaning unity, have lower word scattering. In the other words, on the average, in almost paragraphs of the text, keywords are placed in similarly patterns. Therefore second idea is to propose scattering of the words that placed in the paragraphs. This same idea causes to look at the text as a wave of words. In the other words, such wave as to be draw using to points that are individually indented by a distance to start of text (on the X axis) and a value level equal to every words code (on the Y axis).this means that every word has it's pulse .Pulse levels are not important but also created wave shape and harmony of paragraphs and so total of text are cases that should be care to them. At present this idea to be named Wave Idea.

While signals and waves processing for interpreting a wave or summarize it, we should apply Fourier Conversion on it. Result of this converting is two values as Phase and Domain. A function of these two values is a good criterion for comparing two waves.This idea is implemented in FDS .In this way a text is divided to some paragraphs firstly. Every these paragraphs are calling as a group. In each group, first group's local wave is created. Phase and domain value be extract. The same work will be done on all remain groups. Finally domain and phase scattering of all groups compute by declination criteria.

2 Optimized Stemmer for Persian Language

Are the words such as "ایران", "ایرانی", "ایرانیان" and "ایرانی ها" different?

Although these words maybe are similar by your sight but are different by a text explorer sight. Its reason is so much clear: presence of multi-shape in these words. Then we need a module that can convert the same-stem words to similar form. The name of this module is stemmer. In this project we will use a robust Persian stemmer.

2.1 Implementation of Stemmer

The stemmer has ten phases:
1- Removing "ی" "(/I/) (indefinite article / possessive suffix) 2- Removing auxiliary suffix "ند" /ænd/ 3- Removing possessive and auxiliary suffixes 4- Removing possessive suffixes "ت" /t/ and "تان" /tn/ 5- Removing plural suffixes
6- Removing comparative suffixes 7- Removing other suffixes
8- Removing "ن" /n/ (sign of infinitive) 9- Removing special end letters 10- Removing prefixes

To evaluate this stemmer, a collection of 250 Mb containing 43,680 Farsi documents is used. These documents have several subjects like sport, economic, policy, history, etc.

To evaluate the stemmer 25 queries were used. The relevant documents of each query were selected by a native Farsi speaker.

First, the system (a classic vector-based system) was started up without the stemmer in the indexer and the searcher. The queries were fed to the system and the performance of the system was evaluated (Table 5). Then the system was restarted up using the stemmer, and again it was evaluated with the same queries (Table 6). Comparing these two tables, the system which used the stemmer was 0.151 or 46% better.

3 Implementation of Scoring Method

The best file structure for implementing such model is Inverted List structure. This structure has two main file: words file and information file (or info-file).

There are two columns in the word file .words place in the first column. Second column includes a pointer to info file that indicates where the word is located in it.
Example: ایران → 99200

For searching a question, firstly extract question words. Then for each word reference to info file using word file and compute word locations in each text. Finally ,by attention to what groups each locations is placed ,produce it's signal(wave).and computes similarity value of each paragraphs by using score function.

A most important problem that propounded in recent years is automatic grouping ability. Fast growing in the internet is one of the reasons .many solutions are propounded that is using in internal structure of search engines for increasing their performance. Increasing proportion and similarity of founded documents using by keywords that users input and this has important role in increasing users consent. At present, only way to find application information is manually search and opening the result documents. This work should repeats until you arrive to favorite pages.

Thus in IT industry finding and retrieving documents that relates to keyword is an important problem and many people are working on optimizing information search and retrieve solutions.

When some keywords are searching by a search engine, they are almost relating to one subject. For example when a keyword insert is "peaceful nuclear energy" expects that receive documents relate to nuclear energy and it's peaceful applications.

All mentioned grouping ways return documents that relate to " energy"," nuclear" and "peaceful" .Fourier domain score is useful in order to use document's space information. These information that is more than plain words include location of the word in the text, proportion location of key words together, precedence and priority of words repetition.

In this way location of the words use in (G) vector. Documents that their keywords appear periodically will have more score than documents that their keywords have placed away together. For analyzing relative locations in the text, vectors map to frequency's domain.

Discrete Fourier Conversion able to convert discrete time signals to equivalent frequency. This is actually the same function that need to for converting discrete intervals of text's space words to frequency domain. This process is showed in the figure-1:

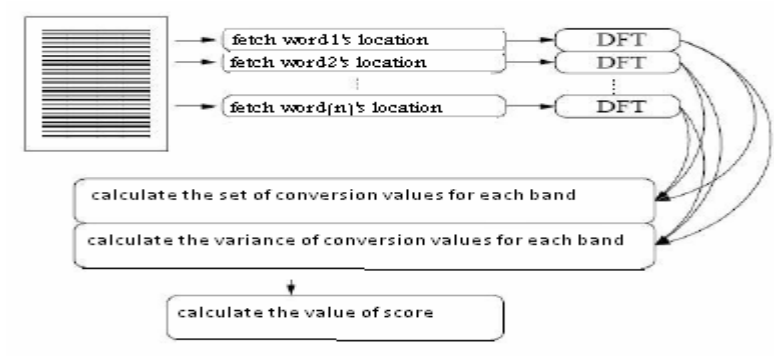


Fig. 1. Fourier Domain Score Method

4 Conclusion

We have designed a new approach which is used to optimize the scoring policy of finding relevant documents in a search engine named ALIA .This Persian search engine is now has been designed and tested as a pilot version. We are working now on this Persian search engine to generate a web application version. Its stemming method would help it to be the best search engine in Farsi language search engines.

References

[1] Shariat, M.J.: Simple Farsi Grammar (Second impression).Asatir, Iran (2000)
[2] Samiei (Gilani), A.: Writing and Editing (Third impression),The Organization for Researching and Composing University Textbooks in the Humanities (SAMT), Iran (2001)
[3] Darrudi, E., Hejazi, M.R., Oroumchian, F.: Assessment of a Modern Farsi Corpus. In: Proceedings of the 2nd Workshop on Information Technology & its Disciplines (WITID), ITRC, Iran (2004)
[4] Taghva, K., Beckley, R., Sadeh, M.: A Stemming Algorithm for the Farsi Language. In: Proceedings of International Conference on Information Technology: Coding and Computing (ITXX 2005), vol. I, pp. 158–162 (2005)
[5] Park, L.A.F., Palaniswami, M., Kotagiri, R.: Internet Document Filtering Using Fourier Domain Scoring
[6] Park, L.A.F., Palaniswami, M., Ramamohanara, K.: Novel Web Text Mining Method Using the Discrete Cosine Transform

FRGA Matching Algorithm in High-Speed Packet Switches

Mohammad Javad Rostami[1] and Ali Asghar Khodaparast[2]

[1] Department of Computer Engineering, Faculty of Computer Engineering
Shahid Bahonar University,Kerman,Iran
mjrostami@mail.uk.ac.ir
[2] Computer Engineering and IT Department
Amirkabir University of Technology, Tehran, Iran
aa_khodaparast@yahoo.com

Abstract. In RG algorithms, each input sends request to all/some of the outputs at each iteration cycle. Then each output sends grant to one of its requesting inputs according to an input priority sequence. Synchronization of input requests or output grants is the reason for further iteration cycles. It may also reduce throughput and cause an unfair service.In this paper, we analyze the existing RG matching algorithms showing that they have fairness and convergence problem under a range of input load distributions. In these years, with increase in data transfer rates and improvement in QoS mechanisms, packet switches really require fair matching algorithms with high convergence speeds. Our analysis and simulation show that our proposed algorithm called FRGA is a high throughput matching algorithm that presents an optimal fairness and convergence under all load distributions.

Keywords: Packet switch,scheduling,matching, convergence.

1 Introduction

Scheduling scheme is the main factor effecting both throughput and fairness in a packet switch. Scheduling scheme depends upon the switch's queuing architecture. The existing architectures mainly include *Shared Memory* [1], *Input Queuing* [2], *Output Queuing* (OQ) [2], *Combined Input and Output Queuing* [3], and *Virtual Output Queuing* (VOQ) [4]. High speed packet switches are usually built by VOQ architecture because of both interconnection simplicity and algorithmic flexibility. Scheduling in this kind of switches is the process of deciding which cells are transferred from inputs to outputs of switch in a timeslot. In fact, the scheduler runs a matching algorithm.

Several matching algorithms have been proposed so far for VOQ switches which can be classified as maximum size and maximum weight matching algorithms [4]. The most popular and practical ones are Request-Grant (RG) matching algorithms. They try to reach a maximal size match in a time-slot. In RG algorithms, each input generally sends request to all/some of the outputs. Then each output grants one of the

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 749–753, 2008.

requesting inputs according to an arbitration policy. There can also be an Accept phase if more than one output can send grant to an input. Since this process may not reach a complete match, it is usually iterated for a few *iteration cycles* or simply *cycles* in a time-slot.

2 The FRGA Matching Algorithm

In this section, we propose a high-throughput RG matching algorithm called *FRGA* (Fair Request-Grant-Accept) which is optimal in both fairness and convergence. The fore-mentioned problems of the existing RG algorithms arise by their sequence update mechanism in time-slot. In fact, we only propose a new sequence update mechanism. FRGA is the same as the general model we discussed in section 1 with the following changes in step 2 and step 3.

2.1 Output Arbiters

If an input is serviced by an output in a time-slot, then the priority sequence of that output arbiter must be updated such that the *N* inputs are fairly ordered in the sequence. Otherwise, we cannot claim a fair service. Let us consider the priority sequence of an output arbiter at the beginning of a time-slot in a *NxN* switch (Fig. 1). In the existing RG algorithms, if the highest priority input (the first input in the sequence) is matched, then the input is moved to the end of the sequence and the others are moved one step to the left (Fig. 1(a)). After the update, the inputs are fairly ordered in the sequence. If the output is matched to the second input because of congestion, then both the first and the second input are moved to the end of the sequence and the others are moved to the left (Fig. 1(b)). This is not fair because the first input is not served. Instead, only the second input must be moved to the end of the sequence (Fig. 1(c)). Therefore, we change the second step of the general model of RG algorithms as follows.

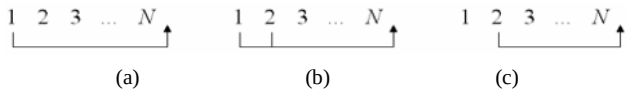


Fig. 1. The behavior of a sequence update mechanism

Step 1. *Grant.* If an output receives any requests, it chooses the one that appears first in a sequence starting from the highest priority input. The output notifies each input whether or not its request was granted. If the granted input accepts the output in the third step of the current cycle, then the input is transferred to the end of the sequence and the inputs located to the right of the granted input are moved one step to the left.

2.2 Input Arbiters

The problem mentioned in the previous section also exists in the sequence update mechanism of input arbiters in the existing RG algorithms. Therefore, we initially change the third step of the general model as follows.

Step 2. *Accept.* If an input receives a grant, it accepts the one that appears first in a sequence starting from the highest priority output. The accepted output is transferred to the end of the sequence and the outputs located to the right of the accepted output are moved one step to the left.

In fact, an input must separate the N outputs into two sets at a time-slot: 1) the outputs for which there is a cell at the input, and 2) the outputs for which there is no cell at the input. Then, the outputs belonging to the second set must get priorities lower than the priorities of the outputs belonging to the first set. Then, the third step of FRGA changes as follows.

Step 3. *Accept.* If an input receives a grant, it accepts the one that appears first in a sequence starting from the highest priority output. The accepted output is transferred to the end of the sequence and the outputs located to the right of the accepted output are moved one step to the left. Then, the outputs for which the input has currently no cell are transferred to the end of the sequence.

3 Simulation Results

In this section, we compare fairness, convergence, and throughput of FRGA and iSlip. Most of the existing RG algorithms operate in the same way of iSlip and no other RG algorithm is required to be compared. We simulated a $N \times N$ packet switch using the iSlip or FRGA matching algorithm under the following two load conditions.

Uniform Load. A 100-percent load at each input with a Bernoulli *i.i.d* arrival process [4].

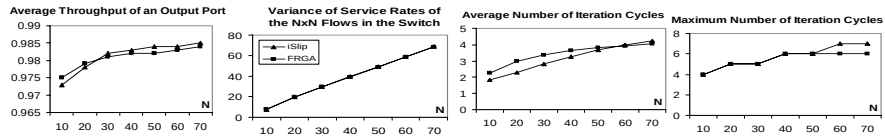
Non-Uniform Load. A 100-percent load with the constant arrival matrix shown in Fig 2. There is a static triangular hole on the matrix and the numbers of 1s are maximally different on the rows. In the case of iSlip, if the situation of output arbiters become synchronous with the hole in a time-slot, then about N cycles are required. As the hole reduces the elements of the output matching array in a time-slot, the output arbiters have priority sequences with different lengths. All the N outputs can try on input 1. There are some outputs that contain only input 1 and a few other inputs in their priority sequences. Hence, when an output with a long sequence tries on input 1, it is probably synchronized with most of the outputs that have small sequences. In this manner, the structure of the arrival matrix helps the output arbiters to become synchronous with the hole.

We compare the two algorithms by the following four parameters:

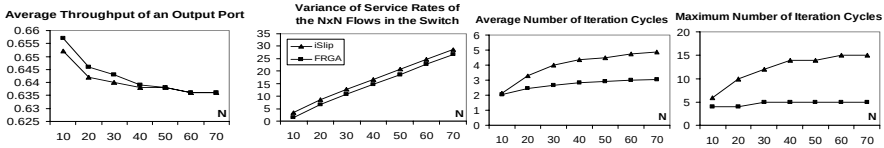
1. Average Throughput of an Output Port: Sum of departure rates (unit: cells per time-slot) of the N outputs of the switch divided by N.
2. Variance of Service Rates (unit: cells per time-slot) of the N^2 flows in the switch: A lower variance shows that the algorithm served the flows by closer rates and hence shows more fairness.
3. Average Number of Iteration Cycles: required by the algorithm to achieve a maximal match in a time-slot.
4. Maximum Number of Iteration Cycles

$$\begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ 1 & 1 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Fig. 2. A triangular non-uniform arrival matrix



(a) Under a uniform load distribution



(b) Under a non-uniform load distribution

Fig. 3. Comparison of iSlip and FRGA in a $N \times N$ switch

iSlip and FRGA under the uniform load are compared in Fig. 3(a). Both iSlip and FRGA are pretty fair and achieve a high throughput. Both the two algorithms requires no more than seven cycles under this uniform load. Although iSlip averagely requires less number of cycles for $N<50$, the average number of cycles required by FRGA is acceptable in a practical packet switch.

iSlip and FRGA under the non-uniform load are compared in Fig. 3(b). Considering the arrival matrix, there is no heavy load towards about half of the N outputs, and this has decreased the average throughput of the outputs. Under this load, FRGA achieves a higher throughput and more fairness than iSlip. Although iSlip and FRGA have no big difference in average number of cycles, the maximum number of cycles required by iSlip is not acceptable in a practical packet switch. With increase in the value of N, maximum number of cycles required by FRGA is rising very slower than that required by iSlip.

4 Conclusion

In this paper, we have analyzed the existing RG matching algorithms and have found that they have fairness and convergence problem under a range of input load distributions. In these years, with increase in data transfer rates and improvement in QoS mechanisms, packet switches really require fair matching algorithms with high convergence speeds. Our analysis and simulation show that FRGA is a high throughput matching algorithm that presents an optimal fairness and convergence in a VOQ switch under all load distributions.

References

[1] Endo, N., Kozaki, T., Ohuchi, T., Kuwahara, H., Gohara, S.: Shared buffer memory switch for an ATM exchange. IEEE Transactions on Communications 41(1), 237–245 (1993)
[2] Awdeh, R.Y., Mouftah, H.T.: Survey of ATM switch architectures. ComputerNetworks & ISDN Systems 27, 1567–1613 (1995)
[3] Chuang, S.-T., Goel, A., McKeown, N., Prabhakar, B.: Matching Output Queueing with a Combined Input Output Queued Switch. IEEE Journal on Selected Areas in Communications 17(6), 1030–1039 (1999)
[4] McKeown, N.: iSLIP: A Scheduling Algorithm for Input-Queued Switches. IEEE Transactions on Networking 7(2) (April 1999)
[5] Anderson, T., Owicki, S., Saxe, J., Thacker, C.: High speed switch scheduling for local area networks. ACM Trans. on Computer Systems, 319–352 (November 1993)
[6] Chao, H.J.: Saturn: a terabit packet switch using Dual Round-Robin. IEEE Communication Magazine 38(12), 78–84 (2000)
[7] Jiang, Y., Hamdi, M.: A 2-stage matching scheduler for a VOQ packet switch architecture. IEEE ICC 2002 4, 2105–2110 (April 2002)
[8] Serpanos, D.N., Antoniadis, P.I.: FIRM: A Class of Distributed Scheduling Algorithms for High-speed ATM Switches with Multiple Input Queues. In: Proc. INFOCOM (2000)
[9] Khodaparast, A., Khorsandi, S.: Design and Analysis of a Fully-Distributed Parallel Packet Switch. In: Proc. APCC 2005 (2005)

A New Class of Data Dissemination Algorithms for Multicast Protocols

Mohammad Javad Rostami[1] and Ali Asghar Khodaparast[2]

[1] Bahonar University, Kerman, Iran
mjrostami@mail.uk.ac.ir
[2] Department of Computer Engineering
Amirkabir University of Technology, Tehran, Iran
aa_khodaparast@yahoo.com

Abstract. In this paper, we show that the existing tree-based or partial-mesh-based data dissemination algorithms do not perform efficiently in traditional customer-provider networks. We propose a new class of data dissemination algorithms called SBDD that considers a full-mesh path structure between multicast nodes. We show that SBDD improves multicast throughput, Latency, fairness, and node dependency over the tree-based or partial-mesh-based algorithms in traditional customer-provider networks. Furthermore, it can also be used in heterogeouse multicast environments with a low computational complexity and a low overhead in comparison with the existing algorithms for such environments.

1 Introduction

A multicast protocol mainly consists of two parts, 1) multicast group management, and 2) data dissemination algorithm. Multicast group management deals with how a node can join/leave the multicast group. Several systems are proposed for this part and one be used in a multicast network depending on system conditions. Data dissemination in peer-to-peer multicast is performed from source node by delivering data to destination nodes over either a tree or a mesh path structure covering all the multicast nodes. In this paper, we consider paths between nodes in such a way that two different paths can contain common physical links on the network. The most important group management systems along with their data dissemination algorithms include Scribe [1], Narada [2], NICE [3], Yoid [4], Gossamer [5], Overcast [6], ALMI [7], Bayeux [8], multicast-CAN [9], SpliteStream [10], Bullet [11], BitTorrent [12], Chainsaw [13], and MeshCast [14].

In tree-based multicast, nodes are organized into a tree-structured overlay with a root at source node. The tree structure defines the routing decisions. In other word, a node receives data from its parent and forwards it to all its children. We divide mesh-based algorithms into two categories, partical-mesh and full-mesh. In partical-mesh-based multicast, multicast nodes are organized into an overlay mesh by selecting multiple nodes as neighbors to exchange data with . In order to enable data exchange between neighbors, the disseminated data is split into smaller data blocks. Each time a node receivers a data block, it informs its neighbors and they can request this missing

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 754–757, 2008.

block. In this paper, we propose SBDD, a class sequence-based data dissemination algorithms, in witch a full-mesh path structure is constructed between multicast nodes. Each node receives one or a few data blocks from the source node and then sends it/them to all the other destination nodes.

Although tree-based algorithms can achieve a near minimized load on the network, they have a few well-known problems [14]. Multi-tree-based and partical-mesh-based algorithms partially solve some of the problems. In this paper, we demonstrate that SBDD achives near-optimal solution for these problems.

The rest of this paper is organized as follows. We present the SBDD class of data dissemination algorithms in Section 2. Section 3 analyzes the performance of these algorithms and Section 4 concludes the paper.

2 Sequence-Based Data Dissemination

In this section, we present a new class of data dissemination algorithms for peer-to-peer multicast. The idea behind these algorithms is to break the multicast data into smaller data blocks and utilize the maximum parallelism and the minimum latency in exchanging the data blocks between multicast nodes.

A. SBDD Data Dissemination Algorithm

There is a source node, *s*, and *N* destination nodes, n_1 , n_2 ,…,n_N . For i = 1 , 2 ,…, N , node n_i shares a upload bandwidth.

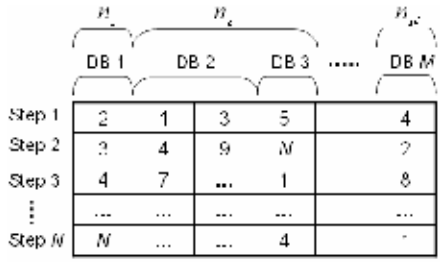


Fig. 1. An example of TSA including *N* nodes and *M* data blocks

We assume that the shared upload bandwidths are normalized to non-zero integers U_1, U_2,...,U_N such that at least one of them equals 1. In fact, these integers represent the relative upload bandwidths of the participating nodes. In the same way, we define D_1, D_2,…,D_N as the normalized download bandwidths of the *N* destination nodes. To simplify calculation, we assume that node *s* have got enough bandwidth and a negligible delay to immediately send the data to the destination nodes, because it has to send the data only one time. The algorithm called SBDD (Sequence-Based Data Dissemination) consists the following steps. *M* is an integer that depends upon network and QoS conditions.

Algorithm SBDD

1. Node *s* splits the multicast data into *M* equal-size blocks.
2. Node *s* calculates as many as *M* integer sequences one per data block.
3. For $i=1,2,\ldots,M$, node *s* assigns data block *i* to a destination node.
4. For $i=1,2,\ldots,M$, node *s* sends data block *i* along with the corresponding integer sequence to the assigned destination node.
5. For $i=1,2,\ldots,N$, node n_i sends each one of its assigned data blocks to all the other destination nodes according to the corresponding integer sequence.

Each destination node has to transfer its assigned data blocks to all the other destination nodes. These data transfers are performed according to the sequences calculated by the source node. In the next section, we show the influences of these sequences on the entire multicast data dissemination and how to generate the sequences under various conditions of network and QoS.

To calculate the sequences, the source node creates an array called TSA (Transfer Sequence Array) (Fig. 1) and fills it by integers including $1,2,\ldots,N$ each one representing the identification number of a destination node. We consider DB as an abbreviation for data block. The rows represent the sequential steps of data transfer. A destination node is related to one column or multiple sequential columns depending on its shared bandwidth. The columns of a destination node are also related to the data blocks assigned to that node. In addition, multiple sequential columns can be related to a data block representing parallel transfer of that block at each step (row) because of the additional bandwidth of the corresponding destination node. If as many as *x* columns of TSA are related to a destination node, then that node has to send *x* data blocks at each step (possibly except the last step) of the data transfer.

3 Performance Analysis

We analyzed how SBDD with TSA behaves by comparison to the existing two models of peer-to-peer multicast data dissemination, tree-based and partial-mesh-based models. Tree-based data dissemination algorithms do not utilize the upload bandwidth of a large fraction of nodes that are leaves in the tree. Multi-tree and partial-mesh algorithms utilize most the upload bandwidths whereas SBDD utilizes all by constructing a full-mesh path structure. In other hand, partially utilizing the upload bandwidths causes shortage in received bandwidth of the destination nodes and thus limits download throughput of them. Therefore, SBDD also utilizes all the download bandwidths of the nodes whereas tree and partial-mesh algorithms do not.

In tree-based algorithms, operation of nodes in the lower level of the tree entirely depends on operation of nodes in the upper level. A data loss in a node or a node departure results in the data stream being lost at all its descendants until the tree is fixed. Partial-mesh approaches partially solve this dependency problem whereas SBDD achieves a minimal dependency such that each node receives data blocks from maximally-different nodes.

4 Conclusions

We showed that the existing tree-based or partial-meshbased data dissemination algorithms do not perform efficiently in traditional customer-provider networks. We

proposed a new class of data dissemination algorithms called SBDD that considers a full-mesh path structure between multicast nodes. We showed that SBDD improves multicast throughput, latency, fairness, and node dependency over the tree-based or partial-mesh-based algorithms in traditional customer-provider networks. Furthermore, it can also be used in heterogeneous multicast environments with a low computational complexity and a low overhead in comparison with the existing algorithms for such environments.

References

[1] Castro, M., Druschel, P., Kermarrec, A.-M., Rowstron, A.: Scribe: A large-scale and decentralized publish-subscribe infrastructure. IEEE Journal on Selected Areas in Communications (JSAC) 20(8), 1489–1499 (2002)

[2] Chu, Y.-H., Rao, S.G., Zhang, H.: A Case for End System Multicast. In: Proc. ACM Sigmetrics (June 2000)

[3] Banerjee, S., Bhattacharjee, B., Kommareddy, C.: Scalable application layer multicast. In: Proc. ACM Sigcomm (August 2002)

[4] Francis, P.: Yoid: Extending the Multicast Internet Architecture (1999), http://www.aciri.org/yoid/

[5] Chawathe, Y.: Scattercast: An Architecture for Internet Broadcast Distribution as an Infrastructure Service, Ph.D. Thesis, University of California, Berkeley (December 2000)

[6] Jannotti, J., Gifford, D., Johnson, K., Kaashoek, M., O'Toole, J.: Overcast: Reliable Multicasting with an Overlay Network. In: Proc. 4th Symposium on Operating Systems Design and Implementation (October 2000)

[7] Pendarakis, D., Shi, S., Verma, D., Waldvogel, M.: ALMI: An Application Level Multicast Infrastructure. In: Proc. 3rd Usenix Symposium on Internet Technologies & Systems (March 2001)

[8] Zhuang, S.Q., Zhao, B.Y., Joseph, A.D., Katz, R., Kubiatowicz, J.: Bayeux: An architecture for scalable and faulttolerant wide-area data dissemination. In: 11th International Workshop on Network and Operating Systems Support for Digital Audio and Video, NOSSDAV 2001 (2001)

[9] Ratnasamy, S., Handley, M., Karp, R., Shenker, S.: Application-level multicast using content-addressable networks. In: Proc. 3rd International Workshop on Networked Group Communication (November 2001)

[10] Castro, M., Druschel, P., Kermarrec, A.-M., Nand, A., Rowstron, A., Singh, A.: SplitStream: high-bandwidth multicast in cooperative environments. In: SOSP 2003: Proceedings of the nineteenth ACM symposium on Operating systems principles, New York, NY, USA, pp. 298–313 (2003)

[11] Kostic, D., Rodriguez, A., Albrecht, J., Vahdat, A.: Bullet: high bandwidth data dissemination using an overlay mesh. In: Proceedings of the nineteenth ACM symposium on Operating systems principles, pp. 282–297 (2003)

[12] Cohen, B.: Incentives build robustness in BitTorrent. In: The 1st Workshop on Economics of Peer-to-Peer Systems, Berkeley, CA, USA (June 2003)

[13] Pai, V.S., Kumar, K., Tamilmani, K., Sambamurthy, V., Mohr, A.E.: Chainsaw: Eliminating trees from overlay multicast. In: IPTPS, pp. 127–140 (2005)

[14] Biskupski, B., Cunningham, R., Dowling, J., Meier, R.: High-Bandwidth Mesh-based Overlay Multicast in Heterogeneous Environments. In: The proceedings of the 2nd International Workshop on Advanced Architectures and Algorithms for Internet Delivery and Applications, Pisa, Italy, October 10 (2006)

Dynamic Point Coverage in Wireless Sensor Networks: A Learning Automata Approach

M. Esnaashari[1] and M. R. Meybodi[1,2]

[1] Soft Computing Laboratory, Computer Engineering and Information Technology Department, Amirkabir University of Technology, Tehran, Iran
{esnaashari,mmeybodi}@aut.ac.ir
[2] Institute for Studies in Theoretical Physics and Mathematics (IPM)
School of Computer Science, Tehran, Iran

Abstract. Dynamic Point coverage in wireless sensor networks is the problem of detecting some moving target points in the area of the network using as little sensor nodes as possible. This can be accomplished by designing a dynamic schedule for making nodes on and off in such a way that in each slice of time, only nodes which can sense the target points in that period are on. In this paper, we propose a novel method for this problem using learning automata. Each node is equipped with a learning automaton which will learn (schedule) the proper on and off times of that node based on the movement nature of a single moving target.

Keywords: Dynamic Point Coverage, Wireless Sensor Network, Learning Automata, Scheduling.

1 Introduction

Point coverage in wireless sensor networks is the problem of detecting some stationary or moving target points in the area of sensor network using as little sensor nodes as possible. This problem can be addressed in many different ways such as designing a deployment strategy which can best address the criterion of minimum number of required nodes [1, 2, 3], rearrangement of nodes assuming movement ability for them [4, 5, 6] and designing a suitable scheduling strategy for making nodes on and off in such a way that in each slice of time (period), only nodes which can sense the target points in that period are on [7, 8, 9]. The last solution is superior to other ones since it can deal with changes occurred in the topology of the network and also prolong the lifetime of the network by allowing only a portion of nodes to be in on state at each slice of time. To our best knowledge, in all decentralized approaches for scheduling, some sort of notification messages are exchanged between nodes in on and off states. Such solutions have two drawbacks; one is the overhead of these notification messages and the other is that nodes in off state should have the ability to receive these messages, and hence they cannot power off their receiving antenna. This leads to high energy consumption even in off periods according to [12].

In this paper, we propose a novel method for addressing the problem of dynamic point coverage in wireless sensor networks using learning automata. This solution can

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 758–762, 2008.

address the two shortcomings of scheduling methods mentioned earlier. This is because in this method, each node (or better the automaton of each node) learns its best on and off schedules using only the information of the moving targets passing through its sensing area, and hence no notification messages exchanged between on and off nodes.

The rest of this paper is organized as follows. The problem statement is given in section 2. In section 3 the proposed method is presented. Simulation results are given in section 4. Section 5 is the conclusion.

2 Problem Statement

We are interested in the dynamic point coverage problem in which a single target point is moving throughout the area of the sensor network and should be detected by nodes which are close enough to it. The problem is to determine the precise times of going to off state and coming back to on for each node based on the movement strategy of the target point. We assume that only one 'on period' and one 'off period' are sufficient for scheduling of each node. More specifically, the target passes the sensing area of node *i* every T_{i1} slices of time for a duration of T_{i2} slices, and the problem at hand is to determine these two time slices for each node in the network. Determination of going to off state is not so complicated. On the other hand, finding the precise time of coming back to on state is more challenging if T_{i1} and T_{i2} change throughout the lifetime of the network.

3 Proposed Method

Each node *i* in the network is equipped with a learning automaton and starts the algorithm with off state period set to 1 time slice. Learning automaton of each node has three actions; *extending* or *shortening* the off state period, and *no change* action. At the beginning, *extending* action has the probability very near to 1 and *shortening* and *no change* actions both have the probability very near to 0. As the algorithm goes on, the off period gradually increases and gets closed to T_{i1}, and hence the probability of *extending* action should decrease to near 0 whereas the probability of *no change* action increases to near 1.

The proposed algorithm is as follows: Each node starts in on state, sensing its surrounding area to determine whether it can sense the target or not. If target cannot be sensed, the node continues in on state until it can sense the target. When the target can be sensed, the node continues in on state, keep on monitoring the target positions. Whenever the node cannot sense the target anymore, it switches to off state, but before that, it determines its off duration using its learning automaton. Learning automaton of the node, selects one of its actions. Based on the selected action, previous off duration will be extended, shortened or remained unchanged. Node goes off for the specified duration and then becomes on. Coming into on state, the node checks for presence of the target for some short duration (*CHECKING_PERIOD*) and based on the result of this checking, rewards or penalizes its learning automaton:

- If the target cannot be sensed, *extending* action is rewarded.
- If the target is sensed right after the node becomes on, *shortening* is rewarded.
- If target can be detected during *CHECKING_PERIOD*, but not immediately when the node comes back to on state, *no change* action is rewarded.

After the above procedure is done, if the target is detected, node continues in on state until the target cannot be sensed anymore and at this time, above procedure is repeated. If the target isn't found in *CHECKING_PERIOD*, node goes off for a very short period and comes back to on state after that, checking for the presence of the target. Going off and coming back to on state is repeated until the target can be detected by the node. At this time, node continues in on state until the target cannot be sensed anymore, then next off period is determined using learning automaton of the node as explained before.

4 Experimental Results

To evaluate the performance of the proposed method several experiments have been conducted and the proposed method is compared with methods given in [7] and [8]. For simulations, a scenario very similar to [9] is used. Assume a road in which target enters from one end and exits from the other end periodically. This is depicted in figure 1. Communication energy estimation is done using the radio model described in [11], and based on the specifications of MEDUSA II sensor node. Binary detection model [10] is assumed for all sensor nodes.

Detection rate, off period rate and energy consumption of nodes are studied for constant and randomly selected speed of the target. Figure 2 gives the results for constant speed and figure 3 gives the results for randomly selected speed.

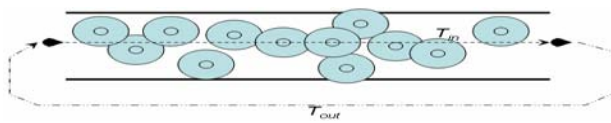


Fig. 1. Simulation scenario: Sensor nodes are scattered randomly through the area of a road and target enters the road from left and exits from right. This sequence is repeated every T_{out} (ms)

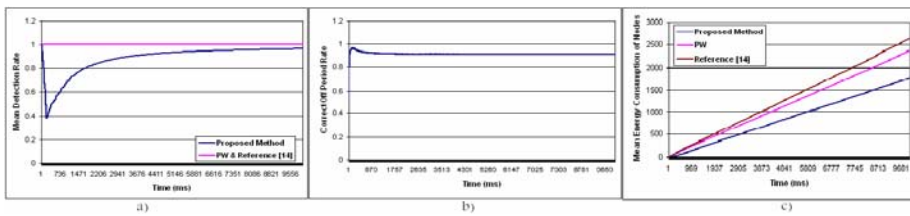


Fig. 2. Detection rate (a), off period rate (b) and consumed energy (c); target speed is constant

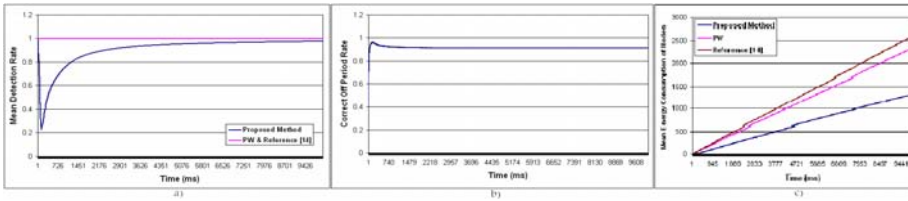


Fig. 3. Detection rate (a), off period rate (b) and consumed energy (c); target speed is random

5 Conclusion

In this paper, we propose a completely different and novel solution to the problem of dynamic point coverage in sensor networks. Previous works in this area use some sort of notification messages which should be exchanged between nodes in on state and nodes in off state. This requires the nodes in off state to have their receiving units on which leads to high energy consumption. In contrast, the proposed method makes use of a learning automaton in each node to make it capable of learning it's off and on periods based on the target path. No notification messages will be required, and hence nodes which are in off state can completely switch their receiving units off. Experimental results show that our method can save much more energy and better prolong the lifetime of the network than other similar methods in the expense of a bit less precision in the detection of the target.

References

1. Dhillon, S.S., Chakrabarty, K., Iyengar, S.S.: Sensor Placement for Grid Coverage under Imprecise Detections. In: Proc. Intl. Conf. on Information Fusion, pp. 1581–1587 (2002)
2. Zou, Y., Chakrabarty, K.: Uncertainty-aware and Coverage-oriented Deployment for Sensor Networks. J. Parallel and Distributed Computing 64, 788–798 (2004)
3. Chen, H., Wu, H., Tzeng, N.-F.: Grid-based Approach for Working Node Selection in Wireless Sensor Networks. In: Proc. of IEEE Intl. Conf. on Communications (2004)
4. Schwager, M., McLurkin, J., Rus, D.: Distributed Coverage Control with Sensory Feedback for Networked Robots. In: Proc. of Robotics: Science and Systems, Philadelphia (2006)
5. Batalin, M.A., Sukhatme, G.S.: Multi-robot Dynamic Coverage of a Planar Bounded Environment. CRES-03-011 (2003)
6. Jung, B.: Cooperative Target Tracking using Mobile Robots. PhD dissertation proposal, University of Southern California (2004)
7. Gupta, R., Das, S.R.: Tracking Moving Targets in a Smart Sensor Network. In: Proc. of IEEE VTC, pp. 3035–3039 (2003)
8. Gui, Ch., Mohapatra, P.: Power Conservation and Quality of Surveillance in Target Tracking Sensor Networks. In: Proc. of the 10th Annual Intl. Conf. on Mobile Computing and Networking, Philadelphia, USA (2004)
9. Jeong, J., Sharafkandi, S., Du, D.H.C.: Energy-aware Scheduling with Quality of Surveillance Guarantee in Wireless Sensor Networks. In: Intl. Conf. on Mobile Computing and Networking, Los Angeles, USA (2006)

10. Chakrabarty, K., Iyengar, S.S., Qi, H., Cho, E.: Grid Coverage for Surveillance and Target Location in Distributed Sensor Networks. IEEE Trans. on Computers 51, 1448–1453 (2002)
11. Koustuv, K.K., Dasgupta, N.P.: An Efficient Clustering-based Heuristic for Data Gathering and Aggregation in Sensor Networks. In: Proc. of the IEEE Wireless Communications and Networking Conf. (2003)
12. Raghunathan, V., Schurgers, C., Park, S., Srivastava, M.B.: Energy-Aware Wireless Sensor Networks. IEEE Singal Processing Magazine (2002-2003)

A Novel Approach in Adaptive Traffic Prediction in Self-sizing Networks Using Wavelets

Hamed Banizaman[1] and Hamid Soltanian-Zadeh[2]

[1] PhD Student of Elec. Eng., Yazd University, Iran
hbanizaman@stu.yazduni.ac.ir
[2] Radiology Image Analysis Lab., Henry Ford Health System, Detroit, MI 48202, USA
[2] Control and Intelligent Proc. Center of Excellence, Dept. of Elec. & Comp. Eng., Univ. of Tehran, Iran
hszadeh@ut.ac.ir

Abstract. In this paper we propose a traffic predictor based on multiresolution decomposition for the adaptive bandwidth control in locally controlled self-sizing networks. A selfsizing network can provide quantitative packet-level QoS to aggregate traffic by allocating link/switch capacity automatically and adaptively using online traffic data. We show that wavelet based adaptive bandwidth control method performs better than other classical methods in the case of average queue size and maximum buffer size. We have compared the performance of different orthonormal wavelets and found that all the other wavelets do far better than Haar with respect to bandwidth utilization factor but Haar shows a very good queue performance. We have studied the effect of other wavelet parameters such as window size, number of decomposition levels and number of filter coefficients. We also introduce a novel adaptive wavelet predictor which can adapt very well to the changes of incoming bursty traffic based on different window sizes and decomposition levels.

Keywords: Wavelet, Self-Sizing, Self-Similarity, DRA.

1 Introduction

Today's multi-service Internet is facing seemingly insurmountable challenge of allocating resources efficiently and to provide guaranteed QoS to an ever increasing network traffic, which is unpredictable, and whose statistical characteristics are unknown. Network researchers have reaffirmed that either capacity overprovisioning or connection level resource reservations (static or dynamic) cannot provide a scalable solution to this problem. A self-sizing network can allocate network capacity automatically and adaptively using on-line traffic data to satisfy the quantitative QoS at packet level. In [6], a self-sizing framework has been proposed for locally controlled networks such as Internet, in which the resource allocation decisions are made at the node level. The main component of the self-sizing network is the algorithm, which is used for bandwidth allocation using predicted network traffic. We need an algorithm which is fast, accurate and robust to provide absolute QoS requirements for aggregate traffic. In [5] number of adaptive bandwidth allocation algorithms have been discussed and Gaussian predictor proposed by Duffield *et al.* was found to be very

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 763–768, 2008.

efficient and best suitable for such applications. Wavelet based traffic models have been proposed by number of authors for different applications. Abry and Veitch have used wavelets to analyze the long-range dependent traffic and have proposed a semi-parametric estimator for the *Hurst* parameter. Multi-fractal wavelet model has been proposed for positive valued data with long range dependent correlations, using Haar wavelet transform. Wang *et al.*, have proposed an adaptive wavelet predictor for modeling VBR video traffic. Xusheng *et al.* use wavelet based models to provide a unified view to include most important understanding of the network traffic in [4]. In this paper we show that our wavelet based traffic predictor is accurate and robust compared to other adaptive bandwidth control algorithms [5]. The technique used in our paper can be applied to online traffic and works at very low time scales. We have improvised the work presented in [3], by analyzing the performance of different orthonormal wavelets. We have also studied the effect of other wavelet parameters such as window size and number of decomposition level. At the end of the article, we introduce a novel adaptive technique which exploits one of these parameters. In the next section we introduce the internet traffic characteristics. We explain signal processing approach to traffic analysis using wavelet in section 3. Experimental results and analysis is given in sections 4, 5 and our conclusions in the last section.

2 Internet Traffic Characteristics

Significant research has been conducted over the last decade analyzing the statistical characteristics of multiplexed data traffic. One of the earliest studies, in [2], led to the examination of such traffic from the perspective of fractal or self-similarity theory. In the case of stochastic objects like time series, self-similarity is used in the distributional sense: when viewed at varying time scales, the object's relational structure remains unchanged. As a result, such a time series exhibits bursts at a wide range of time scales ranging from 10 ms to 100 seconds. Recent work has suggested that the source of such LRD (*long range dependency*) is due to the superposition of many individual On-Off sources.

3 Signal Processing Approach to Traffic Analysis and Prediction

One of the key issues in a measurement-based network rate and congestion control mechanism is the ability to predict bandwidth requirements across a time interval based on current and past measurement data. The objective of this prediction, in the context of real-time control, is to forecast future traffic loads as precisely as possible over a desired time horizon while maintaining reasonable computational complexity.

3.1 Multiresolution Dynamic Resource Allocation Algorithm

DWT consists of decomposition (or analysis) and reconstruction (or synthesis). The DWT captures a signal at various time scales or levels of aggregation. Due to the scale invariance of the basis functions, it is suitable for analyzing properties that are present across a range of time scales, such as LRD.

Consider a data vector, $\hat{X}_k = [X(n-M+1), X(n-M+2), \cdots, X(n)]$ at time n, where k is the time scale and M is an integer. Each element of vector $X(i)$ represents amount of traffic received in time slot i. The algorithm first filters out the DC component in traffic measurements, $\hat{X}$. This DC value is taken as the lower bound for the bandwidth allocation in the next time slot to prevent the application from bandwidth starvation. The signal at the output of the DC filter, $\tilde{X}$, consists of low and high frequency components. The signal is fed into a filter bank in which high pass filter is composed of Haar wavelet coefficients and low pass filter is of Haar wavelet scaling coefficients. The signal $\tilde{X}$ is analyzed by decomposing it into some high frequency sub-bands. The window size M decides the number of sub-band filters used in decomposition. The energy content in each sub-band frequency or time scale k can be computed as:

$$E_k = \sum_{n=2^{k-1}+1}^{2^k} |W(n)|^2, 1 \le k \le K = log_2 M \quad (1)$$

Here we introduce a bandwidth prediction formula according to wavelet energies:

$$BW(n+1) = X_{DC}(n) + \sqrt{\sum_{i=1}^{K} E_i(n)} \quad (2)$$

4 Simulation Experiments, Results and Analysis

We have implemented our prediction-based method in ns-2 and conducted a simulation study to validate the proposed design and compare the performance. To examine the fundamental operation of algorithm within a node (routers ≡ R), a simple topology with a single bottleneck link was used (see Fig. 1). We have used the pareto (on/off) traffic sources to mimic self-similar behavior with the following parameters:

- Number of on/off sources (n) = 100
- Mean on time of a source in millisecond = 0.02
- Mean off time of a source in millisecond = 2.0
- Packet generation rate per second = 500
- Mean inter-arrival time between packets = 0.002

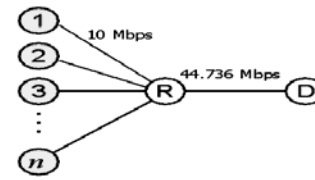


Fig. 1. Simulation topology

In the following sections, we have compared the performance of our predictor with other classic predictors. The comparison is made in terms of *average queue size*, *mean square bandwidth allocation error* (MSBAE) and *maximum buffer size*. MSBAE is the sum of the squares of the difference between each bandwidth demand and the allocation divided by the number of traffic samples. An infinite buffer is assumed, and therefore, no packet loss due to a buffer overflow. We have also studied the performance of different orthonormal wavelets and the effect of other wavelet parameters.

4.1 Comparison of Wavelet Predictor with Other Predictors

In this experiment we have compared the performance of *Average*, *Previous*, *Previous + Average*, *Gaussian* and *Haar Wavelet-based* bandwidth allocation methods. In

Fig. 2, the traffic is shown with vertical axis representing the number of bytes received and horizontal axis the time index with an interval of 0.1 s. According to Table 1, wavelet-energy method is not the best to have a minimum MSBAE. However, it results in the smallest queue and buffer size among the others.

4.2 Comparison of the Performance of Different Wavelet

In this section We have compared the performances of Haar, Symlet (sym4), Daubechies (db4) and Coiflet (coif4) wavelets. Fig. 3 shows predictor performance for different wavelet filters. It is clear that all the other wavelets do far better than Haar with respect to MSBAE but Haar shows a very good queue performance.

4.3 Effect of Window Size on Predictor Performance

Fig. 4 shows average queue size of Haar wavelet predictor with different window size. Also Table 2 shows predictor performance for different window size. It is clear that as we go for higher window sizes, queue and buffer size improves but the error increases.

4.4 Effect of the Decomposition Level on Predictor Performance

In this experiment we have compared the performance of Haar with respect to number of decomposition levels. Fig. 5 shows the queue size performance of wavelet predictor for different decomposition levels. From Table 3 and Fig. 5 it follows that as we go for higher decomposition levels, queue and buffer size improves but the error increases.

5 Adaptive Wavelet Predictor

From the above experiments it can be observed that, the performance of the wavelet predictor can be tuned by the *window size* and *number of decomposition level* used. The basic block diagram for the Adaptive Wavelet Predictor is shown in Fig 6.

5.1 AWP Based on Window Size

In this section, we introduce an Adaptive Wavelet Predictor based on window size. Fig. 7 shows MSBAE of Haar wavelet Predictor for window sizes 2, 4, 8, 16 and 32. We can classify different WS by their margins and adapt window size in real time application according to MSBAE achieved. Table 4 compare AWP performance with Haar wavelet predictor. According to simulation results, window sizes 8 and 4 utilize the most in AWP.

5.2 AWP Based on Decomposition Level

In this section, we introduce an Adaptive Wavelet Predictor based on decomposition level. We use a simple algorithm to compute level for each iteration as follow:

```
dif(n) = MSBAE(n) – MSBAE(n-1);
if dif(n) < 0
      LEVEL(n) = LEVEL(n-1) + 1;
```

```
else
        LEVEL(n) = LEVEL(n-1) - 1;
end
```

Table 5 compare AWP performance with Haar wavelet predictor. According to simulation results, level 2 utilize the most in AWP.

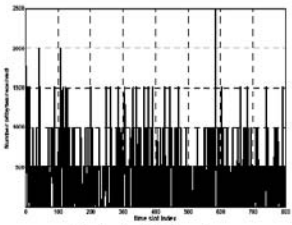


Fig.2. Traffic Samples

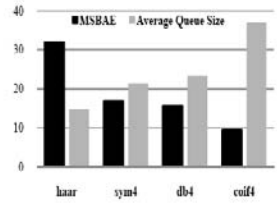


Fig. 3. Comparison of wavelet predictors for different filters, Ws 8

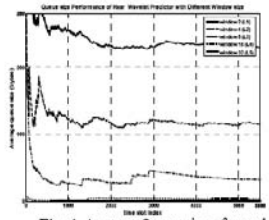


Fig. 4. Average Queue size of wavelet predictors for different window sizes

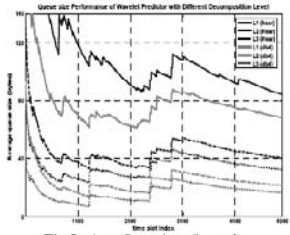


Fig.5. Ave. Que. size of wavelet predictors for different Dec. Level

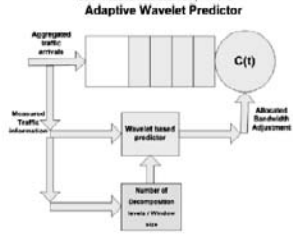


Fig.6. Basic block diagram of Adaptive Wavelet Predictor

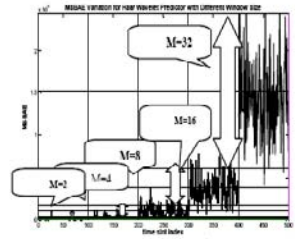


Fig.7 . MSBAE variation for Haar wavel predictor with different window size

Table 1. Comparison of wavelet predictor with other predictors for WS 8

Predictor Type	MSBAE	Ave Que. Size	Max Que. Size
Average	5.41	2215	9187
Previous	7.06	19893	70000
Previous+ Average	8.69	572	5375
Peak	10.8	76	4000
Gaussian	11.8	55	3127
Haar (L2)	11.8	39.7	3000
Haar (L3)	14.8	32	3000

Table 2. Effect of window size on Haar wavelet predictor performance

Predictor Type	WS	MSBAE	Ave. Que. Size	Max Que. Size
Haar (L1)	2	8.97	227	4000
Haar (L2)	4	11.1	114	3500
Haar (L3)	8	14.8	32	3000
Haar (L4)	16	20.6	2.72	1751
Haar (L5)	32	29	0.09	365

Table 3. Effect of decomposition levels for WS

Predictor Type	Level	MSBAE	Ave. Que. Size	Max Que. Size
Haar	1	12.0	80.620	8500
Haar	2	14.0	39.700	3000
Haar	3	14.8	32.016	3000

Table 4. Comparison performance of AWP based on WS, with other Wavelet Predictor

Predictor Type	WS	MSBAE	Ave. Que. Size	Max Que. Size
Haar(L1)	2	8.9	227	4000
Haar(L2)	4	11.1	113	3500
Haar(L3)	8	14.8	32	3000
Haar(L4)	16	20.5	2.7	1751
Haar-Adaptive, *Level= Log M*	*M*	10.9	85.8	4500

Table 5. Comparison performance of AWP based on Dec. Level, with other Wavelet Predictor, WS=8

Predictor Type	Dec. Level	MSBAE
Haar	1	12.076
Haar	2	14.081
Haar	3	14.847
Haar (Adaptive)	Adaptive	13.78

6 Conclusion

We have proposed an adaptive wavelet predictor for locally controlled self-sizing networks. We have compared the performance of wavelet based predictor with other

popular methods. We have done a systematic study of the effect of different wavelet parameters on the predictor performance. We found that there is a trade-off between MSBAE and queue size performance for all wavelet predictors. It is clear that all the other wavelets do far better than Haar with respect to MSBAE but Haar shows a very good queue performance. Our experimental results suggest that wavelet predictor can be made robust by varying the window size and decomposition level adaptively.

References

1. Shafigh, A.: Haar Wavelet prediction-based Fair Queuing. In: Proc. of IEEE ISCN 2006 (2006)
2. Bai, X., Shami, A.: Modeling Self-similar Traffic for Network Simulation. Univ. of Western Ontario (2005)
3. Nalatwad, Srikant: Self-Sizing Techniques for Locally Controlled Networks, PhD thesis, Computer Eng., North Carolina University (2005)
4. Xusheng, T.: A unified framework for understanding network traffic using independent wavelet models. In: Proc. of IEEE INFOCOM 2002 (2002)
5. Siripongwutikorn, P.: A Survey of Adaptive Bandwidth Control Algorithms. IEEE Communications Surveys and Tutorials (2003)
6. Nalatwad, S.: Self-sizing networks: local vs. global control. In: IEEE ICC 2004 (2004)

Simulative Study of Two Fusion Methods for Target Tracking in Wireless Sensor Networks

Saeid Pashazadeh[1] and Mohsen Sharifi[2]

[1,2] Department of Computer Engineering
Iran University of Science and Technology
Tehran, Iran
{pashazadeh,msharifi}@iust.ac.ir

Abstract. Target tracking is one of the popular applications of wireless sensor networks wherein hundreds or thousands of randomly distributed sensor nodes in an environment gather spatio-temporal information from target(s) and send them to a sink node for further processing. Due to various environmental factors on sensor devices, this information is seldom very accurate. Sensor nodes partly process their sensed data using local fusion before sending them to the sink. This paper comparatively studies two major voting algorithms for fusion of target tracking data in intermediate nodes with a view on the accuracy of results. Majority voter and mean voter algorithms are simulated with different densities of sensor nodes to determine the best choice of sensor density for cost effective deployment of sensor nodes. It is shown that formal majority voter yields much more accurate and stable results in location tracking applications than mean voter.

Keywords: Distributed embedded software, non functional requirements, quality of service, wireless sensor network, modeling.

1 Introduction

All computations that use digital devices have errors. There are many different sources causing errors in computations. Specifications of sensors, sensing methods, routing delays and inherent structure of applications are some of the sources of errors in Wireless Sensor Networks (WSNs). These errors are related to the hardware characteristics and sensing methods [1]. Another source of error is computational errors. Inherited errors, cutting errors, and rounding errors account for some of these types of errors. In complex computations, error distribution is an important factor in accuracy of results. For simple computations, we can draw the error distribution tree, which is similar to expression tree, and compute the final error of computation based on accuracy of input data. However, in complex computations, which have stochastic nature, computations of error distributions or formulation of processes is a hard work. An alternative approach would be to simulate these systems.

Local fusion of sensed data within sensor nodes is shown to improve the Quality of Service (QoS) and power saving in WSNs, especially in data centric applications [2]. In object tracking applications that use WSNs, we are faced with thousands of sensor

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 769–772, 2008.

nodes that are randomly distributed in an environment to capture required environmental information and send them to a sink node [3]. In such networks lots of redundant data is sent to middle nodes and sink. Fusion methods try to alleviate this weakness. Many different data fusion algorithms in WSNs have already been proposed. Some of these methods are not appropriate for target tracking applications though which have real-time requirement [4]. One of the applicable fusion methods for tracking applications is the Mean Voter that computes the mean value of spatio-temporal information of target object [5]. Formal Majority Voter is another applicable data fusion method with many variations [5]. We have selected one variation wherein the mean of spatially distributed 3-D vectors of spatio-temporal information of target object is computed first. A nearest vector to the mean vector is chosen as a representative. A vector is than randomly selected from a group of candidate vectors that their Euclidian distance is lower than a specific threshold value σ from the representative vector. In this simulation we assume 0.4 for σ parameter.

2 Simulation Model and Results

We have used the Visualsense simulator that builds on and leverages Ptolemy II [6]. In all simulations we used 30 locator sensor nodes and 10 triangulator sensor nodes with a single sink node. All sensor nodes are spread with normal distribution in 2-D square fields with variation of X position in [100,500] meters range and Y position in [200,400] meters range which gains the density of 500 sensors node per KM^2. All locator sensor nodes are identical, location aware and time synchronized with RF signal range of 100 meters. A moving object sends sound waves in 150 meters effective range every 2 seconds. Locator sensors send the time of sensing the sound of a target object together with its own position to triangulator sensor nodes. Every triangulator sensor node clusters the information that it gets from locator sensor nodes using a sampling time window size. Any triangulator node can compute the location of the target object if it is given the spatio-temporal information from at least 3 different locator sensor nodes.

Simulation was run for 20 steps movement of a target object. Due to the random distribution of sensor nodes, number of collected spatio-temporal data may be more or less in different motion steps of object. Simulation was done with 19 different densities of sensor nodes. For increasing our confidence in results, simulation was done with uniform distribution of sensor nodes with 10 different seed numbers for each individual sensor density. In all simulations we assumed that environment is without noise and our routing algorithm works perfectly without any loss of data and only causes of errors is due to the nature of problem and the deployed fusion method.

Studies show that proper coverage of environment by wireless sensors improves the functionality of WSNs. Given that coverage is dependent on sensor density and the sensing range of sensors, we extensively simulated the two chosen fusion methods in order to compare the effect of sensor density on the accuracy of target tracking under the two methods. In these simulations we computed the mean square error of results.

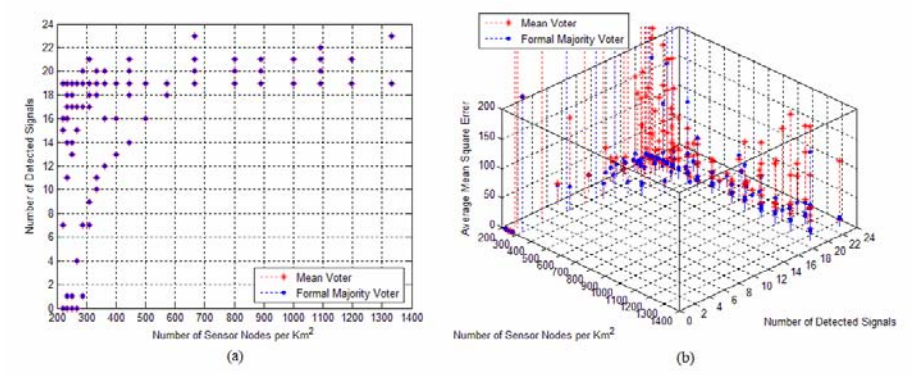


Fig. 1. a. Detected number of signals using two different fusion methods. b. Three dimensional view of simulations results.

Figure 1.a shows the number of signals that can be detected by a WSN in the sink node. The figure shows that the number of detected signals is equal in both fusion methods. This means that the number of detected signals is not dependent on fusion method, but rather depends on the density of sensors in the environment. The figure also shows that in low densities, the number of detected signals is low and more signals of the target are lost. In higher densities, more signals are detected and less data is lost, leading to better functionality of the sink node in monitoring the target object. Figure 1.b shows that the average square error of mean voter in most densities and many runs is higher than formal voter. It shows that the tolerance of average square error in mean voter is higher than the one in the formal voter, so the formal voter is more stable than the mean voter. Figure 1.b shows that the number of detected signals in both voters is around 19. It shows that formal majority voter has more accurate results in different sensor densities.

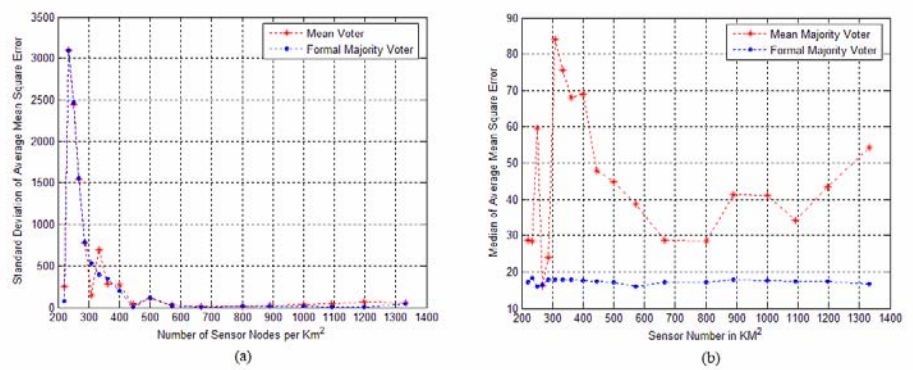


Fig. 2. a.Standard deviation of average mean square error of two different fusion methods. b. Median of average mean square error of two different fusion methods.

Figure 2.a shows that standard deviation of both fusion methods decreases with increases in sensor densities. In addition, it shows that in our typical simulation the density of 600 sensor nodes per KM^2 is a proper choice. Figure 2.b shows the median of average square error of both methods. It shows that formal majority voter is more stable, especially in low densities of sensor nodes, and has lower average square error than mean voter has.

The accuracy of results of formal majority voter is at least twice and sometimes four times than the accuracy of mean voter. Simulation results show that the median of the number of detected signals in both methods is the same and that the results improve with increases in the sensor node densities. As it is somehow shown in all figures, sensor density of 600 nodes per KM^2 is a good choice for our target tracking application with our presumed features. It is thus feasible to carry out a cost-performance analysis on accuracy of results in order to choose a proper density for sensor nodes in tracking applications.

3 Conclusion

Simulation results showed that the formal majority voter has lower inherent error distribution than the mean voter. Average square error of the mean voter is not stable with respect to sensor density. This means that the average square error of simulation runs using the mean voter in different sensor densities varies drastically from one run to another run. This implies that the mean voter uses more redundant data during fusion and thus has less accurate results. On the contrary, the formal majority voter gets results that are more accurate by discarding the results whose Euclidian distances are greater than most of candidate results. It was shown that increases in sensor node densities lead to reductions in errors in results, and that the formal majority voter reaches to a low error rate more quickly than the mean voter does and it requires less density than the mean voter.

References

1. Sharifi, M., Pashazadeh, S.: Using Confidence Degree for Sensor Reading in Wireless Sensor Networks. In: Fourth International Symposium on Mechatronics and its Applications, Sharjah, UAE (March 2007)
2. Liang, B., Liu, Q.: A Data Fusion Approach for Power Saving in Wireless Sensor Networks. In: First International Multi-Symposiums on Computer and Computational Sciences (2006)
3. Yamamoto, H., Ohtsuki, T.: Wireless Sensor Networks with Local Fusion. In: Global Telecommunications Conference, GLOBECOM 2005, vol. 1. IEEE, Los Alamitos (2005)
4. Pullum, L.L.: Software Fault Tolerance – Techniques and Implementation, ch. 7. Artech House (2001) ISBN 1-58053-137-7
5. Broen, R.B.: New Voters for Redundant Systems. Journal of Dynamic Systems, Measurement, and Control (March 1985)
6. Ptolemy (Last visited: August 21, 2007), `http://ptolemy.eecs.berkeley.edu`

The Advantage of Implementing Martin's Noise Reduction Algorithm in Critical Bands Using Wavelet Packet Decomposition and Hilbert Transform

Milad Omidi[1], Nima Derakhshan[1, 2], and Mohammad Hassan Savoji[1]

[1] Electrical Engineering Department, Shahid Beheshti University, Tehran, Iran
[2] Now with Speech Processing group, Iran Univ. of Science and Technology, Tehran, Iran
omidi_sbu@yahoo.com, nima_dr@iust.ac.ir, m-savoji@sbu.ac.ir

Abstract. In this paper we address the problem of enhancing single channel speech signal corrupted with additive background noise. We present a new scheme which utilizes a different time frequency representation along with the psychoacoustic features of human ear and combines these features with the well-known noise estimation method of minimum tracking. Instead of Fourier transform, we use a perceptual wavelet packet decomposition of speech, and perform spectral tracking and filtering on the envelope of the analytic signal.

Keywords: Speech enhancement, Noise estimation, Minimum tracking, Critical band, Hilbert transform, Analytic decomposition, Half-band filter.

1 Introduction

Generally, speech enhancement algorithms consist of a spectral analysis section, followed by noise spectrum estimation, and finally an enhancement filter is applied. Traditionally, Spectral analysis is preformed by calculating the short time Fourier transform. It was observed that in highly nonstationary environments the performance of these algorithms degrades severely. Thus, the topic of noise estimation has recently drawn much attention. A special method of noise estimation was presented by Martin in [1] and he later improved it in [2]. Martin's technique is based on the observation that the power of noisy speech frequently falls to noise power level and by tracking the minima of the noisy signal power we can have an estimate of the noise even during speech segments. Our intention is to implement Martin's algorithm within a different spectral analysis framework. Instead of using DFT filterbanks, we are going to use Perceptual Wavelet Packet Transform (PWPT) as introduced in [3]. Furthermore we utilize squared analytic signal envelope as a representation of subband power. This is based on the experiments in [4] which show that the analytic signal is more reliable than Fourier transform to reveal local variations in nonstatinoary signals.

In the following section we describe the PWPT and analytic decomposition. In section 3 Martin's algorithm is reviewed and next the structure of the speech enhancement system is presented. Experimental results are presented in section 4.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 773–776, 2008.

2 Spectral Analysis

2.1 Perceptual Wavelet Packet Transform

We employ discrete wavelet transform for spectral analysis in our implementation. Wavelet transform is suitable for the study of nonstatinoary processes since it lacks the limitation of fixed size transform window existing in short time Fourier transform. Besides, it was shown that human ear acts similar to a particular structure of filter-bank and if we split the frequency components of the input signal into this structure a better subjective result is achieved. This structure is known as critical bandwidth.

To obtain the critical band structure we must use generalized form of wavelet transform called perceptual wavelet packet transform. 25 critical bands are determined for human ear in the range of 20Hz to 20 KHz. The wavelet decomposition tree is the same as the one employed in [5]. As we assume the input signal to be wideband speech (0-8KHz) the first 19 critical bands are used in our work. In contrast to common methods, we may express a subband by joining a combination of outputs at different levels. To avoid aliasing caused by the speech enhancement section, we must use filters with high selectivity and order. On the other hand, increasing the order of the filter is at the cost of high computational load. Thus there is a tradeoff between filter order and computational load. We have chosen *sav* wavelet kernels [5] due to their sharpness of cutoff. In addition they have the advantage of being linear phase. We use *sav16* as the suitable wavelet kernel with β parameter chosen to be 0.25 for subband decomposition.

2.2 Analytic Decomposition

For every real signal the corresponding analytic signal is formed by suppressing the negative frequencies from the Fourier transform of the original signal. The envelope of analytic signal is of interest since it gives local information of signal, suitable for analysis of nonstatinoary processes. Furthermore, the psychoacoustic model presented in [6], states that a process similar to analytic decomposition is performed in basilar membrane. Among all methods of analytic decomposition, using a half-band filter is most suitable for our purpose. As described in [7], the analytic counterpart of the real signal is obtained by filtering the input signal with a half-band low pass filter which is shifted to right by $\pi/2$. Similar to the suggestion in [5], we use *sav* filter for analytic decomposition with parameters selected to be suitable for this application.

3 Noise Estimation and Noise Reduction

In the original implementation of the minimum tracking algorithm [1] initially a spectral analysis is carried out by using a DFT filterbank. Next, the periodogram of subband is smoothed with the recursive iir filter to be used in the minimum tracking. This algorithm is based on the assumption that the speech signal and the disturbing noise are statistically independent. Besides, it was observed that the power of noisy speech recurrently decays to the power level of noise. Thus, the idea of this technique is to track minima of noisy speech psd to derive noise psd. This minimum tracking is performed in a window long enough to bridge any talk spurts in the speech signal. Since

the minimum is smaller than the average value, this method requires a bias compensation factor to be multiplied to the calculated noise floor.

In our speech enhancement system, we use the tools mentioned in previous section for spectral analysis. A 32ms long segment of input signal X(n) is decomposed to its critical band components using PWPT. The 19 obtained subbands are passed to the analytic decomposition section where the envelope and phase of the subband are extracted. We use squared envelope to represent the short time power of the signal. Consequently, the noise estimate is achieved by applying the smoothed periodogram to the minimum tracking algorithm. To compare with Martin's method we use the simple yet effective filter of spectral subtraction as proposed in [8]. Subsequently, the signal is reconstructed with enhanced envelope and noisy phase. Finally, we calculate the inverse PWPT and use the overlap add algorithm to obtain the enhanced segment.

4 Experimental Results

To perform objective evaluations we used different measures. In addition to common measure of segmental Signal to Noise Ratio Improvement (SNRImp), we also use Log Likelihood Ratio (LLR) and Log Area Ratio (LAR) distance which are based on linear predication analysis [9]. Finally we employ the ITU standard, PESQ [10] (Perceptual Evaluation of Speech Quality) measure to evaluate the quality of enhanced signals. 12 wideband speech signals from 6 male and 6 female speakers were corrupted with white gaussian noise at different SNR values. We applied these sentences to the two noise reduction systems. The results in table 1 show the improvement of enhancement measures. It is observed that the proposed system yields an improvement of LP-based measures and the PESQ score while it fails to improve the SNR measure (less correlated with subjective quality). To test for the capabilities of the system in nonstationary environment, we also applied modulated the gaussian noise as described below:

$$x(i) = s(i) + n(i)\ (\ 1.5 + \sin\ (2\ \pi\ *\ 0.25\ *\ i\ /\ F_s)\ )\ . \qquad (1)$$

Fs is the sampling frequency, *s* is the clean speech, and *n* is white gaussian noise. We use this formula as suggested in [11] to control the amount of nonstationarity of the noise and guarantee that the minimum tracking algorithm performs well for noisy signal.

Table 1. Improvement in SNR , LLR and LAR and the value of PESQ after enhancement

noise type	white gaussian noise			modulated noise		
input SNR	10 db	5 db	3 db	10db	5db	3db
$SNR_{Imp\ proposed}$	0.92	3.83	4.94	0.45	2.79	3.63
$SNR_{Imp\ martin}$	2.92	4.81	5.52	1.04	2.73	3.37
$LLR_{Imp\ proposed}$	2.32	2.25	2.186	0.37	0.47	0.49
$LLR_{Imp\ martin}$	1.48	1.24	1.11	0.13	0.18	0.19
$LAR_{Imp\ proposed}$	1.72	1.64	1.58	0.58	0.63	0.62
$LAR_{Imp\ martin}$	1.01	0.73	0.61	0.29	0.32	0.34
$PESQ_{\ proposed}$	2.90	2.46	2.37	1.79	1.95	2.3
$PESQ_{\ martin}$	2.72	2.44	2.32	1.81	1.93	2.25

References

1. Martin, R.: Spectral Subtraction Based on Minimum Statistics. In: Proc. of EUSIPCO 1994, Edinburgh, U.K, September 13–16, 1994, pp. 1182–1185 (1994)
2. Martin, R.: Noise Power Spectral Density Estimation Based on Optimal Smoothing and Minimum Statistics. IEEE Tran. Speech & Audio Proc. 9(5), 504–512 (2001)
3. Pinter, I.: Perceptual Wavelet Representation of Speech Signals and its Application to Speech Enhancement. Computer Speech and Language 10(1), 1–22 (1996)
4. Derakhshan, N., Savoji, M.H.: Perceptual Speech Enhancement Using a Hilbert Transform Based Time-Frequency Representation of Speech. In: 11th International Conference of Speech and Computer, St. Petersburg, Russia, June 25-29 (2006)
5. Omidi, M., Savoji, M.H.: A New Hilbert Transformer Based on Parametric Wavelet Kernel Application to Analytic Signal Decomposition of Speech Subband. In: 5th International Symposium on Image and Signal Processing and Analysis, ISPA 2007, Istanbul, Turkey (2007)
6. Flanagan, J.L.: Phase vocoder. The Bell System Technical Journal 45, 1493–1509 (1966)
7. Reilly, A., Frazer, G., Boashash, B.: Analytic Signal Generation - Tips and Traps. IEEE Trans. Signal Processing 42(11), 3241–3245 (1994)
8. Berouti, M., Schwartz, R., Makhoul, J.: Enhancement of Speech Corrupted by Acoustic Noise. In: Proc. ICASSP 1979, pp. 208–211 (1979)
9. Hu, Y., Loizou, P.: Evaluation of Objective Measures for Speech Enhancement. In: Proc. of Interspeech 2006, Philadelphia, PA (September 2006)
10. Perceptual Evaluation of Speech Quality (PESQ), an Objective Method for End-to-End Speech Quality Assessment of Narrowband Telephone Networks and Speech Codecs. ITU-T Recommendation, p. 862 (February 2001)
11. Martin, R.: An Efficient Algorithm to Estimate the Instantaneous SNR of Speech Signal. In: EuroSpeech 1993, pp. 1093–1096 (1993)

Real-Time Analysis Process Patterns

Naeem Esfahani, Seyed-Hassan Mirian-Hosseinabadi, and Kamyar Rafati

Computer Engineering Department, Sharif University of Technology, Tehran, Iran
{esfahani@ce,hmirian@sina,rafati@ce}.sharif.edu

Abstract. The influence of Object Oriented Modeling is observable in various areas of software engineering. Embedded and Real-time system domains are not exceptions; several object oriented methodologies have been proposed in these domains. These methodologies have many concepts in common, but the diversity in presentation has concealed their similarities. In this paper, these commonalities in requirement modeling and analysis are captured as process patterns.

Keywords: Real-time system, Analysis, Process Pattern, Embedded system.

1 Introduction

Scheduling, robustness and safety are the main properties of real-time systems. Not paying enough attention to any of these parameters may turn to a disaster. These systems often remain operational for a long period of time and may not be restarted frequently like desktop applications [1].

Software development environments provide little facilities for real-time software development. This makes it obvious that these systems become more bug-prone. So a more thoughtful methodology is essential for managing the production of these systems.

One of the first and most important works performed in late 70's by introduction of MASCOT notation [2]. Later on a design method with a similar name was proposed [3]. Structured analysis and design in real-time development was introduced by RTSAD [4], Ward-Mellor [5] and DARTS [6]. Author of DARTS introduced another methodology named CODARTS [7]; this new methodology integrated proposed concepts from well known methodologies [8], [9,] [10].

Octopus methodology [11], [12] –an object oriented methodology– is based on Jakobson's use case and Rumbaugh's static modeling. It used UML-liked notations. Room methodology [13] is a real-time software design method that depends on a tool named ObjectTime.

After a while, Douglass described how to use UML in real-time software production in his books. In his first book [14], he illustrated how to use UML in modeling real-time systems and in the second one [15], he surveyed the real-time concepts. One of the recent real-time methodologies was COMET [16]. Introduced in 2000, it mainly focused on analysis and design phases. In another research [17], the analysis phase of some real-time methodologies has been studied.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 777–781, 2008.

DESS methodology [18] is another methodology for real-time system development. It uses object oriented in conjunction with component oriented modeling power. This methodology has inherited its concepts from RUP and V-Model. In 2006, STARSS [19] was introduced which is a methodology for fault tolerant real-time system development.

So far, real time embedded methodologies have been reviewed. Ambler, in his two books, [20] and [21], has introduced process patterns. These patterns deal with more general concepts in software development, but less attention is paid to software development lifecycle. The main goal of process patterns is to review the overall lifecycle. In [22] and [23], some notations have been introduced for expressing process patterns.

In this paper, we are going to present the results of our studies about embedded real-time methodologies. We captured the common concepts of these methodologies as a set of process patterns. In the next section, the proposed process patterns are presented. We present the generic requirement modeling and analysis work flow in the third section. Finally, the paper is concluded in the forth section.

2 Process Patterns

Comparing existing methodologies and creating a special purpose methodology requires an overall knowledge. Abstracting existing methodologies helps in gathering required information for achieving this goal. Extracting recurring patterns in methodologies (process patterns) is the key tool in methodology abstraction. These patterns show common features of methodologies. They also show essential activities which should be performed in every methodology. Furthermore, these patterns help us when we are composing a new methodology or configuring an existing one.

2.1 Requirement Elicitation

This pattern (Fig. 1.a) identifies software requirements. It captures software functional model using the use case diagram and use case descriptions. For complex use cases, Alternative flows are introduced. Main flow and alternative ones form events scenarios in the system. The system is considered as a black box.

2.2 Static Modeling

In an iterative fashion the scenarios of the use cases are reviewed and enriched. Required objects and relationships for use case realization are captured. Similar object are grouped into the classes and class relationships are extracted. According to their usage domain, the classes are categorized (Fig. 1.b). In this process following activities are carried out:

- **Identify objects and their relationships:** object(s) that are required for use case realization is identified. Their relationships mainly are based on message passing.
- **Identify classes and their relationships:** similar objects are recognized and their analogous attributes, operation and relationships are classified using classes.
- **Class categorization:** according to their functionality, classes are grouped.

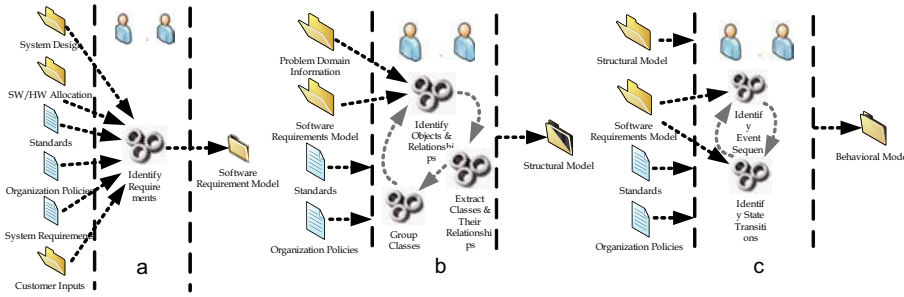


Fig. 1. (a) Requirement elicitation process pattern diagram (b) Static modeling process pattern diagram (c) Dynamic modeling process pattern diagram

2.3 Dynamic Modeling

Focusing on the main scenarios, object communications are captured; the event sequences and state transitions (in different levels) are identified. These sequences and transitions are enriched using alternative flows. It is possible to add new objects and classes to capture all behaviors (Fig. 1.c). This pattern has two activities:

- **Identify event sequences:** communications among object which are captured in static modeling, event sequences and message passing among them are modeled.
- **Identify state transitions:** as a response to the events in the system, there might be some state changes in different levels. These transitions are modeled concurrently with event sequences.

Table 1. The process pattern occurrence in the studied methodologies

	Requirement Modeling	Static Modeling	Dynamic Modeling
Octopus	In the requirement specification phase, after problem definition, use case diagrams and use case sheets are produced. Complicated use cases have transaction scenarios.	In the iterations of the subsystem analysis phase, the class diagram and class specifications are extracted.	After identification of the events, the communication scenarios are determined. The sequence of the events and their results are clarified. For state-driven objects, states are recognized and state diagram is drawn.
DESS	The first step in the realization process is user requirement management. Requirements are stated as business use cases.	Has not defined in this method but it has indicated the structure diagram as output of analysis phase. So it should be prepared somehow.	Again has not been mentioned in this process but the dynamic model is input of the design phase so it must be produced.
STARSS	Pay no attention to this step. It assumes the tasks start with analysis, where we have requirements.	At the beginning of the iterations for generating one level, static model is built. The objects of that level are identified; they will be classified as classes. Relationships are modeled.	At each level, behavioral model is constructed using state transition diagram. Non state driven behavior is modeled by extracting events and their dependencies. In this way all the valid sequences will be identified.
RT-UML	The system context diagram is created. The messages and events between system and environment are characterized. The use case diagram is produced. The scenario for each use case is extracted.	After understanding the outside world and system's relationship with the world, we step in the system and identify objects, classes and their relationships. We also use scenarios and use cases in objects identification process.	Using scenarios, the intra object relations are modeled. Object has a state diagram which determines their behavior. The way objects interact determines what portion of the state diagram will be used in each scenario.
COMET	The system is modeled as a black box. Relationships among the system and the actors are stated in a narrative way. The relationships are defined in use case diagrams.	The physical and the entity classes are more important. After identifying classes we classify them in the groups and subsystems.	The inter- and intra-object behavior is modeled iteratively. Objects' internals are modeled using state diagrams (if object has state driven behavior). Inter-objects behavior is modeled using collaboration diagrams.

2.4 Process Patterns in the Methodologies

In Table 1, we have shown how the proposed process patterns are related to the studied methodologies.

3 Generic Workflow

To complete our patterns, the order and relationship among them should be illustrated. This arrangement could be accomplished by a closer look at the inputs and outputs of the patterns.

In this section, we present a generic workflow for requirement and problem domain modeling. This generic workflow is extracted from the studied methodologies and the indicated order is the same in all of them. Also other arrangements are possible [17].

As depicted in Fig. 2, requirements are modeled in the requirement modeling phase. Next, the models are analyzed; first the static model of the domain is built and the result is further completed by adding behavior in dynamic modeling. Static and dynamic modeling are performed iteratively; The result is improved gradually.

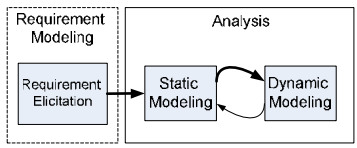


Fig. 2. Generic workflow

4 Conclusion

The focus of this paper is on requirement modeling and analysis of embedded real time systems. The existing real time embedded methodologies have employed diverse mechanisms for this purpose, but they have followed analogous policies. We gathered and presented their similarities as three process patterns.

The general purpose object oriented methodologies have concepts in common with real-time methodologies studied, but significant differences exist in their point of view to the problem domain; e.g. modeling non functional requirements, especially time, is more crucial. Interactive nature of real-time systems leads us to a special kind of system decomposition: system is viewed as several enveloping layers. In these systems, controller plays a critical role; so, state driven analysis becomes important for supporting history dependent decision making.

Currently we are continuing our research to elaborate the next phases, particularly the design phase. Preliminary results show that real-time nature of these methodologies is even more significant in design and implementation. As a future work, we are going to introduce a generic lifecycle for real-time object oriented methodologies.

References

1. Kopetz, H.L.: Software engineering for real-time: a roadmap. In: IEEE Software Engineering Conference, pp. 201–211. IEEE, Ireland (2000)
2. Simpson, H., Jackson, K.: Process Synchronization in MASCOT. The Computer Journal 17(4) (1979)
3. Simpson, H.: The MASCOT Method. IEE/BCS Software Engineering Journal 1(3), 103–120 (1986)
4. Hately, D., Pirbhai, I.: Strategies for Real Time System Specification. Dorset House, New York (1988)
5. Ward, P., Mellor, S.: Structured Development for Real-Time Systems, vol. 1, 2, 3. Yourdan Press, New York (1985)
6. Gomaa, H.: A software Design Method for Real-time Systems. Communications, 938–949 (1984)
7. Gomaa, H.: Software Design Method for Concurrent and Real-time Systems. Addison-Wesley, Reading (1993)
8. Jackson, M.: System Development. Prentice Hall, Englewood Cliffs (1983)
9. Parnas, D., Clements, P., Weiss, D.: The Modular Structure of Complex Systems. In: 7th IEEE International Conference on Software Engineering, Orlando, Fla. (1984)
10. Booch, G.: Object Oriented Design with Applications. Addison-Wesley, Reading (1991)
11. Awad, M., Kuusela, J., Ziegler, J.: Object-oriented Technology for Real-time Systems: A Practical Approach Using OMT and Fusion. Prentice Hall, Englewood Cliffs (1996)
12. Ziegler, L.: Applying object-oriented technology in real-time systems with the OCTOPUS method. In: ICECCS 1995, p. 306. IEEE, Los Alamitos (1995)
13. Selic, B., Gullekson, G., Ward, P.: Real-Time Object-Oriented Modeling. Wiley, New York (1994)
14. Douglass, B.P.: Real-time UML, 2nd edn. Addison-Wesley, Reading (1999b)
15. Douglass, B.P.: Doing Hard Time: UML, Objects, Frameworks, and Patterns in Real-time Software Development. Addison-Wesley, Reading (1999a)
16. Gomaa, H.: Designing Concurrent, Distributed, and Real-time Applications with UML. Addison-Wesley, New York (2000)
17. Kimour, M.T., Meslati, D.: Deriving objects from use cases in real-time embedded systems. Information & Software Technology 47(8), 533–541 (2005)
18. Van Baelen, S., Gorinsek, J., Wills, A.: The DESS Methodology. Version 01 - Public, Deliverable D.1 (2001)
19. Kan, P.: The STARSS Methodology, Department of Computer Science, King's College London, Technical Report TR-06-03 (2006)
20. Ambler, S.W.: Process Patterns: Building Large-Scale Systems Using Object Technology. Cambridge University Press, Cambridge (1998)
21. Ambler, S.W.: More Process Patterns: Delivering Large-Scale Systems Using Object Technology. Cambridge University Press, Cambridge (1999)
22. Gnatz, M., Marschall, F., Popp, G., Rausch, A., Schwerin, W.: Towards a Living Software Development Process Based on Process Patterns. In: 8th European Workshop on Software Process Technology (2001)
23. Störrle, H.: Describing Process Patterns with UML. In: 8th European Workshop on Software Process Technology (2001)

Selecting Informative Genes from Microarray Dataset Using Fuzzy Relational Clustering

Soudeh Kasiri-Bidhendi and Saeed Shiry Ghidary

Department of Computer Engineering, Amirkabir University of Technology, Tehran, Iran
{kasiri,shiry}@aut.ac.ir

Abstract. Selecting informative genes from microarray experiments is one of the most important data analysis steps for deciphering biological information imbedded in such experiments. This paper presents a novel approach for selecting informative genes in two steps. First, fuzzy relational clustering is used to cluster co-expressed genes and select genes that express differently in distinct sample conditions. Second, Support Vector Machine Recursive Feature Elimination (SVM-RFE) method is applied to rank genes. The proposed method is tested on cancer datasets for cancer classification. The results show that the proposed feature selection method selects better subset of genes than the original SVM-RFE does and improves the classification accuracy.

Keywords: Gene selection, microarray data, redundancy reduction.

1 Introduction

The DNA microarray technology is emerging recently in the field of computational biology [1]. Classification based on microarray data faces with many challenges. The main challenge is the overwhelming number of genes compared to the number of available training samples, and many genes are not relevant to the distinction of samples. Gene selection is a process that selects a small subset of genes from the full set, prior to data classification [1]. Gene selection problem can be broadly divided into two categories: Gene ranking and gene-subset evaluation. Gene ranking involves a criterion function for measuring the discriminative power of individual genes. Some examples of criterion function are: TNoM score [2] and Park score [3]. This ranking is simple but it has three drawbacks: (i) there is not any prior knowledge to determine the size of the subset. (ii) Selected genes may be redundant. (iii) Ranking considers only the individual gene discriminative ability and the combined effect of genes is ignored [4].

Combining two low ranked genes may obtain higher discriminative information than combining two high ranked genes. There are attempts to minimize the redundancy [5] by measuring pairwise gene correlation within the selected set. Gene-subset evaluation methods have subsequently been proposed to overcome those drawbacks. The most informative subset must be found by performing greedy search to evaluate all possible gene combinations [6] or stochastic search [7] is often an alternative choice for obtaining a sub-optimal solution. In [8], linear SVMs are used in a backward

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 782–785, 2008.

elimination procedure for gene selection, and the selection procedure is referred to as SVM recursive feature elimination (SVM-RFE). Compared with other feature selection methods, SVM-RFE is a scalable, efficient wrappers method.

In this paper fuzzy relational clustering (FRC) is used for redundancy reduction and eliminating genes with similar expressions. Then SVM-RFE is used to rank genes and select the most informative genes. This method is tested on cancer classification tasks based on gene expression data. The remainder of the paper is organized as follows: In section II, proposed method is described. In Section III, numerical experiments on publicly available colon and Leukemia cancer datasets are reported. Finally, conclusion remarks are presented in Section IV.

2 Proposed Method

The first part of proposed method, fuzzy relational clustering, is used to eliminate redundancy of gene expression. Only one gene from each cluster would be selected as informative gene. FRC uses cosine distance, and clusters genes having similar expressions in samples. Then SVM-RFE is used as ranking criterion to select informative genes.

2.1 Fuzzy Relational Clustering

Traditional fuzzy relational clustering (FRC) can be summarized as follows [9]:

- Determine the set of samples to be clustered. Let $X = \{x_1, x_2, ..., x_m\}$ be a set of data where x_i is a $1 \times N$ vector with real values.
- Establish the fuzzy similarity matrix: to construct the fuzzy similarity matrix R, the first step is to calculate the similarity indices $r_{ij} = R(x_i, x_j)$ of x_i and x_j where r_{ij} can be any arbitrary similarity function.
- Transform fuzzy similarity matrix, R, into a fuzzy equivalence matrix $\mathbf{R}^*$=matrix $(r_{ij}^*)_{i,j=1..m}$: A fuzzy similarity matrix of size $m \times m$ should be composed by itself at most $m-1$ times to be converted to a fuzzy equivalence matrix.
- Calculate λ-cut matrix of. The λ-cut matrix $\mathbf{R}^*$ can be defined as follows:

$$r_{ij}^{\lambda} = \begin{cases} 1 & r_{ij}^* \geq \lambda \\ 0 & r_{ij}^* < \lambda \end{cases} \tag{1}$$

Similar rows of matrix R_{λ}^* form a cluster. With an equivalent matrix and different thresholds, different clustering results will be obtained. However, in the case of microarray with high dimensions, fuzzy equivalence matrix calculation is computationally too complex to be considered practically. This problem can be solved by finding connected components in an undirected graph [10].

2.2 SVM Recursive Feature Elimination

SVM-RFE feature selection method was proposed in [8] to conduct gene selection for cancer classification. Nested subsets of features are selected in a sequential backward elimination manner, which starts with all the feature variables and removes one feature variable at a time. At each step, the coefficients of the weight vector of a linear SVM are used to compute the feature ranking score.

3 Experiments

The proposed method is evaluated on Colon and Leukemia cancer dataset. The expression data in raw format are available at [11] and [12] respectively. Classification accuracy of applying proposed and SVM-RFE methods on datasets is shown in Table 1. Fig. 3 shows the test errors of linear SVM classifiers on gene subsets selected respectively by SVM-RFE and proposed method. As shown in table 1 and fig. 3 using fuzzy relational clustering for omitting redundancy of co-expressed genes leads to better performance with fewer genes. The fuzzy relational clustering does not need a pre-determined number of clusters. In particular, by choosing different values for λ, different number of clusters and different cluster shapes may be acquired. Last but not least, it's worth noting that clusters resulted from applying other methods have hyper-spherical shapes which may not be the case in many problems while clusters resulted from applying Fuzzy relational clustering do not have restriction. Therefore, genes that express differently in distinct sample conditions are selected for ranking with SVM-RFE and it improves the classification accuracy with fewer genes.

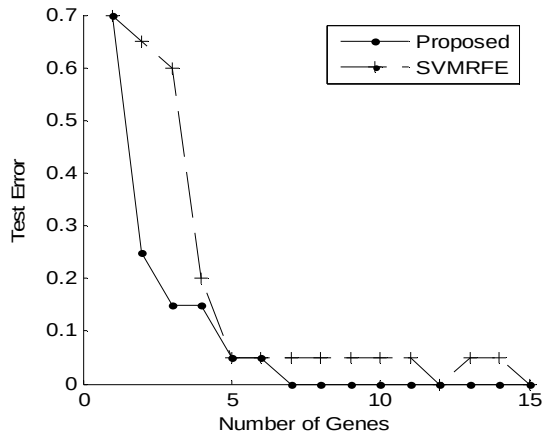


Fig. 1. Performance of feature subsets selected by SVM-RFE and Proposed Method on Colon dataset

Table 1. Results of experiments

Method	Dataset	Number of Genes	Train Accuracy	Test Accuracy
SVM-RFE	Colon	7	100%	95%
Proposed	Colon	7	100%	100%
SVM-RFE	Leukemia	4	100%	97%
Proposed	Leukemia	4	100%	100%

4 Conclusion

Gene selection plays an important role in analysis of microarray datasets. In this paper, fuzzy relational clustering is used to cluster co-expressed genes and select genes expressed differently in distinct sample conditions. Then Support Vector Machine Recursive Feature Elimination method is applied to rank genes. We conclude that the proposed method can select better gene subsets than SVM-RFE and improve the cancer classification accuracy.

References

1. Manfred, N., Laiwan, C.: Informative Gene Discovery for Cancer Classification from Microarray Expression Data. In: IEEE Workshop on Machine Learning for Signal Processing, pp. 393–398 (2005)
2. Ben-Dor, A., Bruhn, L., Friedman, N., Nachman, I., Schummer, M., Yakhini, Z.: Tissue Classification with Gene Expression Profiles. Journal of Computational Biology 7, 559–584 (2000)
3. Park, P.J., Pagano, M., Bonetti, M.: A Nonparametric Scoring Algorithm for Identifying Informative Genes from Microarray Data. In: Proceedings of the Pacific Symposium on Biocomputing, pp. 52–63 (2001)
4. Guyon, I., Elisseeff, A.: An Introduction to Variable and Feature Selection. Journal of Machine Learning Research (3), 1157–1182 (2003)
5. Wang, M., Wu, P., Xia, S.: Improving Performance of Gene Selection by Unsupervised Learning. In: Proceedings of Neural Networks and Signal Processing, vol. 1, pp. 45–48 (2003)
6. Inza, I., Sierra, B., Blanco, R.: Gene Selection by Sequential Search Wrapper Approaches in Microarray Cancer Class Prediction. Journal of Intelligent and Fuzzy Systems 12, 25–34 (2002)
7. Deutsch, J.M.: Evolutionary Algorithms for Finding Optimal Gene Sets in Microarray Prediction. Bioinformatics 19, 45–52 (2003)
8. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection, for cancer classification using support vector machines. Machine Learning 46, 389–422 (2002)
9. Bojadziev, G., Bojadziev, M.: Fuzzy Sets, Fuzzy Logic, Applications. World Scientific, New Jersey (1995)
10. Dong, Y., Zhuang, Y., Chen, K., Taib, X.: A hierarchical clustering algorithm based on fuzzy graph connectedness. Fuzzy Sets and Systems 157, 1760–1774 (2006)
11. `http://www.broad.mit.edu/cancer`
12. `http://microarray.princeton.edu/oncology`

GMTM: A Grid Transaction Management Model

Ali A. Safaei, Mostafa S. Haghjoo, and Mohammad Ghalambor

Department of Computer Engineering, Iran University of Science and Technology
Tehran, Iran
{safaeei,haghjoom,ghalambor}@iust.ac.ir

Abstract. A grid database is a collection of logically interrelated databases (distributed over a grid environment) which is heterogonous, autonomous and contains replica. Special treatments in activities such as transaction management are necessary. In such an environment, each site itself should handle requests, collaborate with others, and follow up transactions. Despite importance of transaction management aspect in grid databases, no appropriate model is proposed yet. In this paper a transaction management model is presented for grid databases. It has a vast range of decision making and scheduling constructs to support all types of transaction activities with high performance. Activities such as forwarding, delegation, conditional commit, and compensation are supported.

1 Introduction

Grid Database is a collection of one or more databases logically interrelated (distributed over a grid environment) which can also be heterogonous, autonomous and contain replica [1]. Therefore, distinct autonomous distributed heterogonous databases cooperate to reach grid DB goals. A well-known grid database architecture is OGSA-DAI which is a middleware for implementation of the GGF DAIS specifications. It consists of both a framework and a tool to provide a uniform interface to access disparate, distributed, heterogeneous data sources in grid environment.

A grid DB must offer an efficient, robust, intelligent, transparent, uniform access to data. So, many problems appear to researchers in term of architecture, query processing, preserving data consistency, transaction management, etc. Transaction management in grid is one of the most important aspects with no appropriate proposed model for it, yet. In this paper a transaction management model named “Grid Mega Transaction Model (GMTM)” is presented. It has a vast range of decision making and scheduling constructs to support all types of transaction activities with high performance. Activities such as forwarding, delegation, conditional commit, and compensation are supported.

2 Related Work

In order to preserve compatibility to existing distributed systems, a transaction processing architecture based on the OGSA platform and the X/Open DTP model is proposed in

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 786–789, 2008.

[2].To provide a formal method for grid transaction processing, in [3] a preliminary theoretical model called Membrane Calculus based on membrane computing and Petri nets is proposed. In [4] grid DBMS concept which is a system for dynamically managing data sources in grid environments is presented. Requirements, architecture and required services to have grid-DBMS are introduced.

3 The Transaction Management Model

In contrast to ordinary distributed database systems, the absence of a site as controller or coordinator in grid database systems causes plenty of problems. In such an environment, each site itself should handle requests, collaborate with others, and follow up transactions. We present a model to satisfy such requirements. Due to realities of grid environments, we have to borrows some traditional concepts (e.g. compensation), but we adapt and customize them to the new environment and add necessary constructs.

3.1 The Transaction Manager Architecture

Our grid transaction manager architecture consists of two main components: ***decomposition & planning*** and ***submition & completion control*** (figure 1).

The decomposition and planning component decomposes grid mega transactions (user requests) into subtransactions to be executed in some sites. The best site for execution of each subtransaction is found. The collection of subtransactions and information about the best sites to execute them is sent to the submition and completion control component. Three cases may happen:

- the local site itself is able to perform some subtransactions, the submition and completion control component gives them to the local DBMS to **execute**.
- the local site is not able to perform some other subtransactions, but finds some sites (named *friend*s) which have this capability. It **forwards** them to those sites.
- some parts of transactions may be incomprehensible neither in the local site nor in any friends. They are **delegated** to possible (*foreign*) sites which have more information about them, or ignore if no such sites are found.

We classify four categories of *schedulers* as transaction trees to support decision making and automation:

- ***serial grid transaction*****:** Is a transaction tree in which subtransactions are submitted and committed sequentially so that one is submitted after commitment of the previous one. Such subtransactions are dependent to each other sequentially and may use results provided by previous ones.
- ***parallel grid transaction*****:** allows all its subtransactions to be submitted and committed in parallel. All committed ones get aborted if one of them aborts.
- ***serial-alternative grid transaction*****:** In cases where there are some acceptable options with preference, this type is used. It is a transaction tree in which subtransactions are submitted one after another until one of them succeeds. The transaction aborts only if the last child aborts.

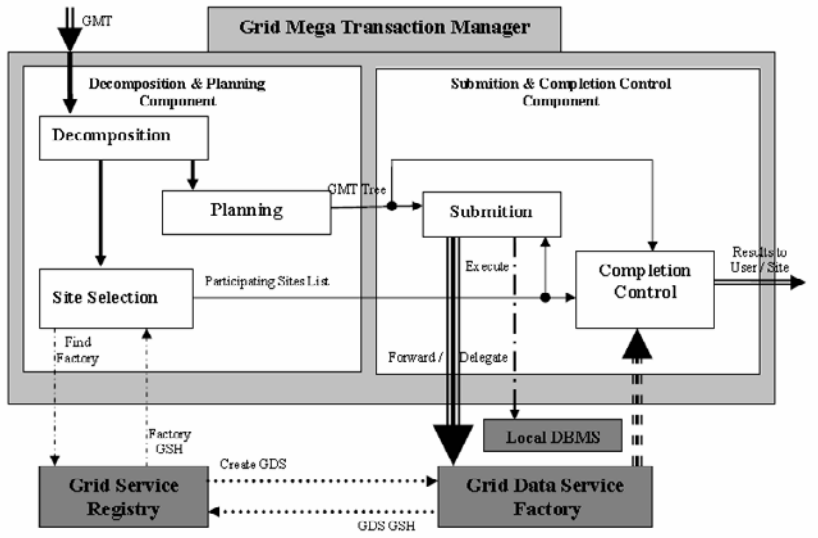


Fig. 1. Grid mega transaction manager architecture

- ***parallel-alternative grid transaction*:** Is a transaction tree in which all subtransactions are submitted in parallel to get a quick result. The transaction commits as soon as one of them is committed. Others get aborted immediately.

As said before, some parts of a request might be incomprehensible to the local site and no friends are found to handle them. Such parts are **delegated** to some sites if possible. Delegation means that the new site has full jurisdiction over that request and is responsible for it. It may execute, forward, decompose and plan or even delegate the transaction (or parts of it). The intermediate sites are omitted and the result is directly sent back to the original site.

A **grid mega transaction tree** is a tree in which: the root is a grid mega transaction. Leaves are executable tasks, which are either atomic or compensable. Internal nodes are either schedulers or delegation with sub trees at foreign sites. Sub-trees are finally decomposed into schedulers and/or leaves.

3.2 Transaction Processing Algorithm

Each request is passed to the ***decomposition and planning component.*** It is decomposed into some subtransactions by the *decomposition unit* and the collection is sent to *site selection unit* which classifies subtransactions into the following classes: ***a)*** executable in the local database system ***b)*** forwarded to a *friend* site ***c)*** subtransactions which are incomprehensive or out of activity field of the local and friend sites. They are delegated to a *foreign* site. Albeit it is possible that the target site can't perform the request either, but it might know some other sites to do that. In order to find participating sites to perform subtransactions which are not executable in the local database systems, the site selection unit consults *Grid Service Registry* (GSR) described in [5] as a client with the mechanism. It sends a request including required services for

subtransactions to the GSR. GSR returns GSH of service of factories for each subtransaction which can be performed in a friend site. For incomprehensive parts, the GSR may introduce more than one site from which the unit selects the best one as the foreign site and delegates the subtransaction to it. In parallel with site selection, the *planning unit* forms received collection of subtransactions into a grid mega transaction tree.

The tree and the list of target sites are given to the ***submition & completion control component***. It is composed of *submition unit* and *completion control unit*. Submition unit submits subtransactions to the determined participating sites according to the GMT tree. Completion control unit makes inquiries about subtransactions completion to control the tasks and returns final results to the user or requesting site.

4 Conclusion and Future Research

In this paper a transaction model named "*Grid Mega Transaction Model (GMTM)*" is presented. It observes user's criterions and tries to accept all possible types of requests, using novel scheduling and planning mechanisms. Local autonomy preserving plus optimum performance for both user's request and other concurrent transactions are its main advantages. We plan to work on concurrency control and recovery in near future.

References

[1] Aloisio, G., et al.: The Grid-DBMS: Towards Dynamic Data Management in Grid Environments. In: Proceedings of IEEE ITCC 2005 (2005)
[2] Qi, Z., et al.: Integrating X/Open DTP into Grid Services for Grid Transaction Processing. In: Proceedings of the 10th IEEE FTDCS 2004 (2004)
[3] Qi, Z.: Membrane Calculus: a formal method for Grid transactions. Concurrency Control and Computation: Practice and Experience (2006)
[4] Goel, S., Hema Sharda, T.: Atomic commitment and resilience in grid database systems. Int. J. Grid and Utility Computing 1(1) (2005)
[5] Grid Database Service Specification Primer, GFD-I, Global Grid Forum (2000)

Design of a Custom Packet Switching Engine for Network Applications

Mostafa E. Salehi, Sied Mehdi Fakhraie, Abbas Banaiyan, and Abbas Hormati

School of Electrical and Computer Engineering,
University of Tehran Tehran 14395-515, Iran
{mersali,fakhraie,banaiyan,hormati}@ut.ac.ir

Abstract. This paper addresses the design of an application-specific processor with emphasis on packet processing applications. In this domain, demands for increasing the performance and the ongoing development of network protocols both call for flexible and performance-optimized engines. We present a flexible engine that is optimized for packet switching operations. The architecture and instruction set of our packet switching engine is designed based on a quantitative study of packet switching applications. Based on the extracted guidelines, architecture of our proposed communication micro engine (CME) is designed and verified with functional simulation. Comparative results show that at least 2-3 times performance gain can be achieved by using CME. A single CME core operates at 400Mbps, while it has an area of 1.6 mm^2 and consumes 60mW of power when synthesized in 0.13µm CMOS technology.

Keywords: Aplication Specific Processor, Packet Switching Engine, and Packet Processing.

1 Introduction

Evolution of computer networks, along with the ever increasing users' demand for improved networking services, has imposed the development of high-capacity telecom systems. Studies have shown that the link bandwidths supported by these systems have doubled every year [1]. The rapid evolution of optical networking technology has shifted the network bottleneck to the execution of various networking tasks. As a result, performance has been a primary focus of network processing elements (PE), and there has been tremendous interest in speeding these modules.

PEs are traditionally either ASICs or general purpose processors (GPPs). GPPs are preferred due to their flexibility while ASICs provide better performance. Today, network processors (NPs) offer both flexibility and high performance. However, GPPs will still be used for initializing, configuring and orchestrating the NP control path [2][3][4][5].

Network processors are high-performance and programmable embedded processors designed to implement complex packet processing tasks at high line speeds. Today's commercial network processors are optimized for packet header processing by deploying custom processor engines. The Intel IXP1250 [6] uses six micro-engines, IBM PowerNP [3] uses 8 processing units, Motorola C-5 [4] uses 16 processing units,

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 790–794, 2008.

and GigaNetIC [7] exploits configurable number of processing elements in a scalable parallel SoC architecture. This implies using special purpose processors optimized for specific tasks. These special purpose processors must be flexible and also are supposed to work in a speed close to an ASIC solution.

In this paper we first use the results of two GPPs handling the entire tasks of a layer-2 switching application. We use these results to offer an optimized ASIP architecture and instruction set for networking tasks in our ongoing work. The rest of this paper is organized as follows. Section 2 describes the performance challenges of switching applications and extracts architectural guidelines for a packet switching engine. In Sections 3 and 4 we propose the architecture and instruction set of our switching engine, and Section 5 presents the performance and synthesis results of our proposed communication micro engine (CME).

2 Extracted Architectural Guidelines

Our benchmark for extraction of architectural guidelines is layer-2 switching (L2S) application 8]. We have implemented the entire layer-2 switching task on both LEON2 and PowerPC processors. The working environment is a Xilinx FPGA-based system and hardware development platform. It has one Xilinx V2P50 FPGA9], four 100Mbps MACs operating in full-duplex mode and one SRAM memory, which is used as frame storage.

As shown in [8], the most frequent instructions of the L2S program are shift, logical and memory instructions. The shift and logical instructions are used in bit-manipulation operations and checking of the branch conditions. Therefore, a CPU with dedicated bit-manipulation instructions could be helpful for this application. For example a single bye update instruction, can substitude five instructions in the SPARC assembly code. Our bit-manipulation instructions include set/clear or test a bit position and also update a specific byte of a 32-bit word. These instructions are used for checking the branch conditions, header field extraction, and preparing the command words of the custom hardware blocks such as lookup tables.

According to [8], instructions that access to header data through the bus are only 0.02% of the total instructions; however, these instructions consume 26% of the total execution time. Thus, a typical RISC processor is I/O bound in header processing and a special technique is required to relieve the processing core from I/O duties 11]12][13]. According to the spatial locality of the header data, we propose some specific load and store instructions for accessing the packet header.

A proper instruction cache can also optimize the bus transactions. An instruction cache reduces the number of single load instructions to the bus memory. As shown in [8], the instruction cache leads to 40% optimizations in the packet processing time.

Accesses to the memory on bus result in high latencies, but much of the latency can be overlapped with local computation. The memory overlap can exploit a simple out of order execution scheme, and give a chance to the proceeding instructions after the load to be executed in the memory response waiting period.

3 Proposed Architecture and Instruction Set for CME

In order to have a high performance communication micro engine (CME), we suggest techniques to optimize bus access latency by reducing the bus access overhead and also organizing the accesses to the bus. We also propose a data manipulation unit (DMU) for reducing the computational time of bit-manipulation operations. Our proposed solutions are as follows:

Developing dedicated instructions for reducing computational time of the bit-manipulation operations. DMU performs the sequence of instructions performing bitwise operations in a single cycle bit manipulation instruction.

Developing dedicated instructions to reduce bus access time. The proposed burst load/store instructions can be used to read the entire header at once, thus reducing single bus accesses and improving the overall performance.

Developing an internal vector interrupt controller. External interrupts are directly connected to the VIC engine within our processor. This will help the processor to quickly find out about the source of interrupt.

Developing a non-blocking memory access by implementing a simple out of order instruction execution scheme. This technique allows the instructions proceeding a blocked memory instruction to be fetched and executed, unless these new instructions are themselves memory instructions or they need the memory result.

Developing multi-layer bus architecture. This technique can reduce the bus arbitration time when there are many request for accessing the bus.

We have implemented the proposed improvements in the CME, with a pipeline of six stages. The datapath of CME is shown in Fig. 1.

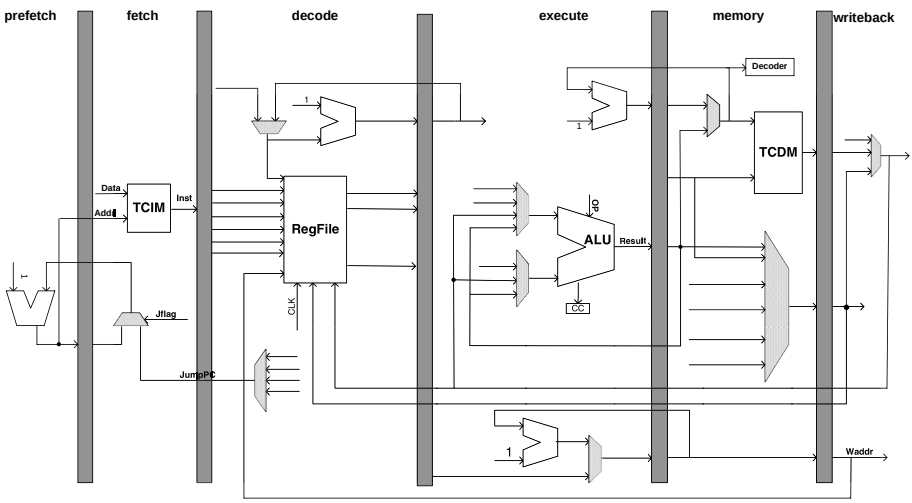


Fig. 1. CME datapath

For instruction set design, we use a combination of general and special purpose instructions. General-purpose instructions are used to give more flexibility for any further changes in switching flow and special purpose instructions are used to boost the execution of the packet processing tasks and therefore increasing our switching performance. The instruction set is designed based on the application requirements, quantified in terms of the required micro-operations and their frequencies.

4 CME Performance and Synthesis Results

Taking advantage of our instruction-level profiler, and exploiting the emulation results, we have measured the overall switching time and system performance. Fig. 2 shows the total switching capacity of various packets on LEON2, PowerPC, and CME processors using 3:1 CPU and bus clock frequency ratio (i.e. 240MHz CPU clock frequency and 80MHz bus clock frequency).

Considering the 672-bit length (minimum frame size), 280Kframe/s can be processed with LEON2 processor, and 208Kframe/s frames can be processed with PowerPC processor. With this processing power almost one 100Mbps (148Kframe/s) line can be processed. However, our goal is to provide adequate processing time in SOHO applications, with the target of four 100Mbps Ethernet ports. In order to achieve the throughput of 400Mbps, the processing time of a minimum size frame should be less than 1.7µs. According to the obtained results, we can reach the required performance of 400Mbps switching capacity with CME working in 280MHz.

CME is synthesized using a 0.13 micron CMOS library, considering slow library with slow operating condition. (voltage: 1.08v and temperature: 125). According to the obtained results, the clock frequency of CME is 300MHz, the total area is 1.6 mm^2 while its power consumption is about 60mW, 85% of the chip area is occupied with register file and memory modules, and these modules consume about 70% of the chip power consumption.

CME is a single-core embedded processor for home gateways, small office/home office (SOHO) routers, and other networked embedded applications. Fig. 3 compares the power consumption of CME with some network processors that have similar performance demands.

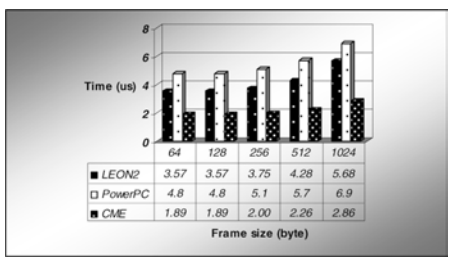


Fig. 2. Total switching time of L2S on LEON2, PowerPC, and CME

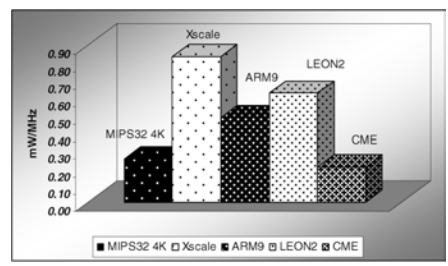


Fig. 3. Power/Frequency ratio for some embedded processors used in network processors *(results are in 0.13 microntechnology)*.

As shown in Fig. 3, CME has acceptable power consumption in similar performance demands. IXP420[14] combines a high-performance Intel Xscale[17] processor, ADM512015] consists of a high performance embedded MIPS4K CPU, and Conexant 16] exploits ARM9 processors for packet processing application.

References

1. Mudigonda, J., Vin, H.M., Yavatkar, R.: Overcoming the memory wall in packet processing: hammers or ladders? In: Proc. of symposium on Architecture for networking and communications systems, pp. 1–10 (2005)
2. Intel Corporation, IXP1200 Network Processor Datasheet (December 2001), http://www.intel.com/design/network/datashts/278298.htm
3. Allen, J., Bass, B., et al.: IBM PowerNP network processor: Hardware, software, and applications. IBM Journal of Research and Development 47(2/3), 177–194 (2003)
4. Freescale Semiconductor, Inc., Motorola C-5 Network Processor Data Sheet, Silicon Revision B0 (2005), http://www.datasheetcatalog.com/motorola/17/
5. Vlachosa, K., Orphanoudakis, T., et al.: Design and performance evaluation of a Programmable Packet Processing Engine (PPE) suitable for high-speed network processors units. Microprocessors & Microsystems 31(3), 188–199 (2007)
6. Intel Corporation, IXP1250 Network Processor Datasheet (December 2001), http://www.intel.com/design/network/datashts/278371.htm
7. Niemann, J.C., Puttmann, C., et al.: Resource efficiency of the GigaNetIC chip multiprocessor architecture. Journal of Systems Architecture 53(5–6), 285–299 (2007)
8. Salehi, M.E., Rafati, R., Baharvand, F., Fakhraie, S.M.: A quantitative study on layer-2 packet processing on a general purpose processor. In: Proc. of International Conference on Microelectronic, pp. 218–221 (2006)
9. Xilinx, Silicon Devices, Virtex-II Pro FPGAs (2006), http://www.xilinx.com/products/siliconsolutions/fpgas/virtex/virtex_ii_pro_fpgas/index.htm
10. Mudigonda, J., Vin, H.M., Yavatkar, R.: Managing memory access latency in packet processing. ACM SIGMETRICS Performance Evaluation Review 33(1), 396–397 (2005)
11. Serpanos, D.N.: Speeding up high- speed protocol processors. IEEE Computer 37(9), 108–111 (2004)
12. Pnevmatikatos, D.N., Sourdis, I., Vlachos, K.: An efficient, low-cost I/O subsystem for network processors. IEEE Design & Test 20(04), 56–64 (2003)
13. Papaefstathiou, I., Vlachos, K., et al.: Packet processing acceleration with a 3-Stage programmable pipeline engine. IEEE Communications Letters 8(3), 183–185 (2004)
14. Intel, IXP420 network processor, product brief, http://www.intel.com/design/network/prodbrf/252494.htm
15. Infineon product information, http://www.infineon.com/cms/en/product/channel.html?channel=ff80808112ab681d0112ab68d6500057
16. Conexant products, http://www.conexant.com/products/
17. Contreras, G., Martonosi, M.: Power prediction for Intel XScaler processors using performance monitoring unit events. In: Proc. of the international symposium on Low power electronics and design, pp. 221–226 (2005)

Multiple Robots Tasks Allocation: An Auction-Based Approach Using Dynamic-Domain RRT

Ali Nasri Nazif, Ehsan Iranmanesh, and Ali Mohades

Department of Math and Computer Science,
Amirkabir University, Tehran, Iran
{nasri,iranmanesh,mohades}@aut.ac.ir

Abstract. Consider a 2D environment, in which multiple robots, initially at random configurations, can move. We propose an auction-based method for the allocation of tasks to these robots. We consider tasks as some points in the environment. The environment is occupied with static obstacles, but other moving robots can be considered as dynamic obstacles and may prevent a robot from reaching its goal. So rebidding may be necessary at definite times.

Our method is based on Dynamic-Domain RRT, which is suitable for cluttered environments and narrow passages. On the other hand, we will see that our method does not need a roadmap, which encompasses all regions of the environment and so reduces the time complexity.

Keywords: Motion Planning, Task Allocation, Dynamic-Domain RRT, Single Auction.

1 Introduction

This paper addresses the problem in which multiple robots are required to do a set of tasks that assigned to them. Using multiple robots instead of one, can reduce the total time needed to accomplish the tasks. In this problem, we should consider two main issues: first of all, we should mention a method for distributing tasks among robots efficiently and the second one is to plan trajectories for each of the robots, a path that is collision free with static obstacles and also other moving robots that act just like dynamic obstacles in the environment.

For the first subject, means distribution of tasks among robots, auctions have been suggested for many years [1].

We have two kinds of auctions: combinatorial and single. Combinatorial auctions, in which all the robots can bid on bundles of tasks instead of just one task [2], may be complex and confusing and may need high computation. So we use single-item auction, in which just one item at a time is auctioned and the robots bid on that sole item and do not confuse themselves for a group of tasks [3, 4].

In this paper we improve the method used in [4] that tries to shorten the total time taken to complete all the tasks. It is flexible and let rebidding whenever necessary.

For the second subject, means building trajectories for robots from initial configuration to goal configuration, we use Dynamic Domain RRT [5], which shows significant improvement above existing RRT-based planners.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 795–798, 2008.

2 Framework

In the next section we explain distribution of task among robots via sequential single-item auction and the bidding strategies we use in our method and in section 2.2, we describe our way of planning trajectories for multiple robots.

2.1 Allocation of Tasks Via Auction

We use sequential single-item auction for distributing of tasks among our robots. The robots are at random positions at the beginning of the method and we consider the robots as points that can move in the free configuration space. We also model the tasks as locations (points) in our environment outside the obstacles. We consider an external robot as the auctioneer, which can communicate with other robots. At the beginning of the method, this robot is given the entire set of tasks. This robot then acts as an auctioneer and distributes tasks one at a time based on a first-price auction.

We consider some time limitation on our method. Here tasks have priorities that represent the tasks importance. The tasks are auctioned based on their priorities, starting with the highest priority task.

Tasks with no offer are discarded at the moment until next rebidding of tasks. There is no guarantee for accomplishing all the tasks because of the considered time limitation and we want to accomplish the tasks with highest priorities as much as possible.

For the auction to be done, we need an estimation of the distance between the current position of each robot and the auctioned task. Because as you will see in the next section, we consider robots as dynamic obstacles, the best estimation to use is just a direct line between the start and the goal position of each robot. After the auction for one of the tasks performed, the winner robot starts to move to the location of that task.

We consider five kinds of tasks: free, assigned, under-execution, pending and accomplished.

Free tasks are those which are not assigned to any robots yet. Assigned tasks are those that the auctioneer has performed auction on them, but the winner robots have not achieved them yet. Accomplished tasks are those which the winner robots have done them and any further operations are not needed on them. Under-execution tasks are those which the assigned robots to them are now operating on them and still have not been accomplished and pending tasks are the tasks which have been pended in order of higher priorities tasks as we will explain.

As our purpose is accomplishing tasks with higher priorities, when a robot detects a higher priority task than the currently assigned task, if there is no free robot for assigning that task, that robot postpone execution of its currently assigned task to complete the higher priority one first. It means that the free robot with the best bid will win the auction and if there is no such a robot, the robot with the best offer, is forced to delay its current task in favor of another task with a higher priority. Later, the postponed task can be executed from its delayed time by the same robot or any other robots. This is because of considering all of the robots homogeneous.

For each of the winner robots, we consider an expected time to reach to its goal means the assigned task position based on the direct distance between its current and

goal positions. The rebidding is necessary in two situations: first when a task has been accomplished and second when the expected time for a robot to reach to its goal is expired.

Now consider an example in figure 1. We consider 3 tasks *T1*, *T2* and *T3* and 3 robots *R1*, *R2* and *R3*. The index numbers show the priorities of tasks. For example we suppose *T3* has a higher priority than *T2*. We suppose that the task *T1* is under execution while the robot R3 accomplished the task *T3* and at that time the robot R2 is on its way to the task *T2*. Because of a task accomplishment, a re-auction is necessary. The re-auctioning is done on the task which is free or assigned and has the higher priority among other free or assigned tasks. So the re-auctioning is done on the task *T2*. Because the expected time related to *R2* has not expired until now, although the robot *R1* should win the auction, the task should be assigned to *R2* until it has time to reach its goal. This causes that our algorithm not to be so confusing with consideration of many pending tasks. But if the expected time finished, a re-auction is done, but this time the robot *R1* postpone its current time because of a task with higher priority and moves toward the task *T2*. The postponed task can be accomplished later by *R1* or any other robots.

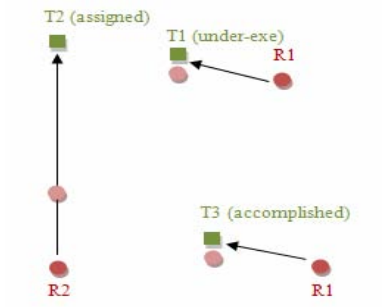


Fig. 1. Three different kinds of tasks; the red circles show the start positions of the robots and the light red circles show the current positions of robots

2.2 Making Trajectories

The robots follow the paths generated by the RRT- connect method using Dynamic-Domain. In previous works such as [3], a roadmap has been used to encompass all regions of the environment like a Rapidly Random Tree that explores unknown regions of spaces. After making the roadmap, they try to find a path that connects start and goal configurations on this roadmap using a method such as Dijkstra.

In many cases, we do not need to explore all unknown regions of the environment and all we need is just connecting the current location of the winner robot to its assigned task location. Because of that, we use RRT-connect which tries to connect two RRTs that rooted at the start and goal configurations. On the other hand, previously proposed methods had problems with narrow corridors and in many cases supposed goals on such situations as unreachable configurations. By controlling the sampling domain using Dynamic-Domain RRT we can handle such problems.

You can see a comparison between these two methods in the following figure:

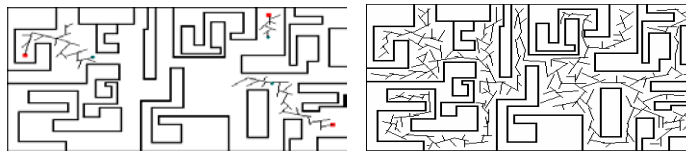


Fig. 2. Comparing the two mentioned strategies

References

1. Sutherland, L.E.: A futures market in computer time. Comm. of the ACM 11(6), 449–451 (1968)
2. Berhault, M., Huang, H., Keskinocak, P., Koenig, S., Elmaghraby, W., Griffin, P., Kleywegt, A.: Robots Exploration with Combinatorial auctions (2002)
3. Nanjanath, M., Gini, M.: Dynamic Task Allocation for Robots via Auctions (2006)
4. Gerkey, B.P., Matarić, M.J.: Sold!: Auction methods for multi-robot coordination. IEEE Trans. on Robotics and Automation 18(5) (2002)
5. Yershova, A., Jaillet, L., Simeon, T., LaValle, S.: Dynamic-Domain RRTs: Efficient Exploration by Controlling the Sampling Domain (2005)

Efficient Computation of N-S Equation with Free Surface Flow Around an ACV on ShirazUCFD Grid

Seyyed Mehdi Sheikhalishahi[1], Davood Alizadehrad[2], GholamHossein Dastghaibyfard[3], Mohammad Mehdi Alishahi[4], and Amir Hossein Nikseresht[5]

[1] Graduate Student
alishahi@cse.shirazu.ac.ir
[2] Graduate Student
alizadehrad_davood@yahoo.com
[3] Assistant Professor, Department of Computer Science & Eng., College of Eng., Shiraz University, Shiraz, Iran
dstghaib@shirazu.ac.ir
[4] Professor, Department of Mechanical Eng., College of Eng., Shiraz University, Shiraz, Iran
alisha@shirazu.ac.ir
[5] Assistant Professor, Department of Mechanical Eng., Shiraz University of Technology, Shiraz, Iran
nikser@sutech.ac.ir

Abstract. This paper presents the application of a parallel high accuracy simulation code for Incompressible Navier–Stokes solution with free surface flow around an ACV (Air Cushion Vehicle) on ShirazUCFD Grid environment. The parallel finite volume code is developed for incompressible Navier–Stokes solver on general curvilinear coordinates system for modeling free surface flows. A single set of dimensionless equations is derived to handle both liquid and air phases in viscous incompressible free surface flow in general curvilinear coordinates. The volume of fluid (VOF) method with lagrangian propagation in computational domain for modeling the free surface flow is implemented. The parallelization approach uses a domain decomposition method for the subdivision of the numerical grid, the SPMD program model and MPICH-G2 as the message passing environment is used to obtain a portable application.

Keywords: Grid Computing, MPICH-G2, Computational Fluid Dynamic, Air Cushion Vehicle, Navier-Stokes, Load Balancing.

1 Introduction

From the point of view of numerical modeling [9], the solution of mathematical physics problems is sufficient challenge task. In the recent years the high-speed maritime industry has experienced significant changes in the development process of new vehicles, one of which is hovercraft. The generic title 'hovercraft' is used to cover the two principal types of marine vehicles discussed in the literature, namely Air Cushion Vehicle (ACV) [1] and Surface Effect Ship (SES) [10, 11]. The present day, ACV

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 799–802, 2008.

has, in principle, no part of the craft in contact with the land or sea surface and is therefore completely amphibious. Air is supplied from a lift fan, which provides airflow round the periphery of the hull followed by ejection into the cushion space. The hydrodynamic aspects of an air-cushion vehicle(ACV) has been studied by assuming its action to be equivalent to that of a pressure distribution acting on the free surface of the water. This idealization prohibits any physical contact of the lower edge of the craft with the water. In addition, it has been assumed that the flow of escaping air under the periphery is inviscid, and therefore produces no spray. In this study, primarily we have developed a parallel CFD [2] application for ACV simulations. We also devise a heuristic dynamic load balancing algorithm in heterogeneous environments such as Grid [3]. Finally, we evaluate performance of different MPI implementations. In section 2 briefly comes issues related to implementation of ACV simulation and results appear in section 3 and finally the last section provides the conclusion.

2 Implementation

At the present study the domain has been decomposed into the vertical strips [4] because of the direction of flow field is horizontal and as it is mentioned in the previous articles, when using of line-by-line method, choosing vertical lines results in losing convergence time. Special effort had to be taken out for the parallelization of the higher order differencing schemes such as QUICK, where the use of a 5 point stencil in one dimension for the discretization requires an additional cell layer for the IPB, in order to supply correct local data exchange. CFD simulations on heterogeneous systems present several major challenges in order to achieve effective load balancing. The load balancing process and consequent data movement must be very fast in comparison to the overall run-time. According to the results and good speedup of ACV CFD simulation parallelization on homogenous systems, a heuristic algorithm for dynamic load balancing of problem implemented. The algorithm does its activity in the first iterations of problem to provide load balancing for next iterations.

3 Experiments and Results

The ShirazUCFD Grid setup between the HPC Center in the Mechanical Engineering department and the Grid Computing laboratory in the Computer Science and Engineering department at Shiraz University. Two centers are connected by the Shiraz University optical fiber that operates at 100 Mbits per second. Currently, the ShirazUCFD Grid comprises two dedicated Linux clusters. The clusters run the Globus Toolkit [5] version 4.0.3 as Grid middleware and Torque [6] version 2.1.3 as local resource manager. Specifications of clusters that build the ShirazUCFD Grid are listed in Table 1. In addition, in each cluster, nodes are connected through their Local Area Network that operates 100 Mbits per second. Since our network connection is fairly homogenous, good results are anticipated from this Grid test bed.

ACV CFD simulation jobs have been submitted to cross-site and single-site by MPICH-G2 [7] middleware to compare their results. Cross-site communications and single-site communications are based on the MPICH-G2 library. The calculations for ACV CFD simulation are carried out for two different grid sizes, 209*164(34276 points) and 296*210(62160 points). For single-site runs, ACV simulations have been submitted to

HPC Cluster head node. Fig. 1 shows the speedup ratio of parallelization with respect to the number of processors on the HPC cluster. As seen in Fig. 1, the speedup is quite good. The speedup increases with the number of grid points and decreases with the number of processors. In cross-site runs (up to 8 processors) half of the processors are chosen from Grid Computing laboratory and the other half from HPC Cluster. For tests that need more than 8 processors, 4 processors come from Grid computing laboratory and the rest from HPC Cluster. The ShirazUCFD Grid is not very heterogeneous; therefore speedup ratio can be a good performance evaluation measurement in this scenario.

Table 1. Specification of ShirazUCFD Grid resources

Name	Site	Nodes	CPU	RAM	Role
Server	HPC	1	Intel® 2.8GHz	1GB	Head
cnode1-cnode11	HPC	11	Intel® 2.4GHz	1GB	Compute
Persepolis	Grid Lab.	1	Intel®3.4GHz	2GB	Head/Compute
gnode1-gnode3	Grid Lab.	3	Intel®2.8GHz	512MB	Compute

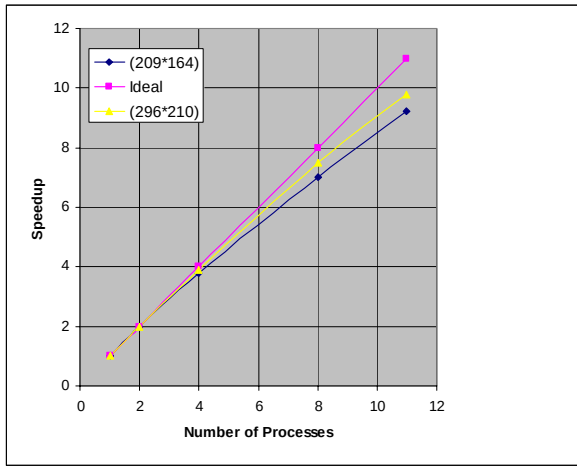


Fig. 1. Speedup of the calculations on the HPC cluster nodes

Using two processors as basis, one with 2.4GHz speed from HPC Cluster and the other with 2.8GHz speed from Grid computing laboratory, the speedup ratio was estimated in cross-site runs. It is seen from Fig. 2 that the dynamic load balancing scales well.

One of the other contributions of this paper is a comparison of the grid-enabled MPI (i.e. MPICH-G2) with MPICH-2.1.0 and MPICH-1.2.7 that are the cluster implementation of MPI [8]. To compare the performance of different MPI implementations, jobs are submitted to HPC cluster. According to the results of different experiments, MPICH-G2 outperforms the other MPIs. Furthermore, MPICH-2.1.0 outperforms MPICH-1.2.7. For instance, for a job which took 11:23 hours using MPICH-G2, it took 13:07 hours using MPICH-2.1.0 and 12:34 hours using MPICH-1.2.7. Some of the more important reasons that lead to these results are as follow:

- First, MPICH-G2 jobs use Globus Toolkit's services for data management (GASS). This is one of the main reasons because CFD simulations generate a vast amount of output.
- Second, MPICH-G2 jobs by using Globus Toolkit libraries for communication do better latency handling in networks. They select the best communication method for intra-machine messaging.

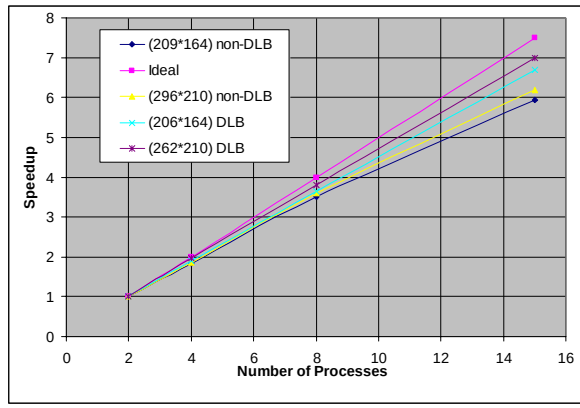


Fig. 2. Speedup of the calculations on the ShirazUGrid environment

References

1. Doctors, L.J.: The Forces on Air Cushion Vehicle Executing an Unsteady Motion. In: 9th Symp., Naval Hyd., Paris Proc., O. N.R., Wash. D.C (1972)
2. Versteeg, H.K., Malalasekekea, W.: An Introduction to Computational Fluid Dynamics: The Finite Volume Method. Longman Group Ltd. (1995)
3. Foster, I., Kesselman, C., Tuecke, S.: The anatomy of the Grid: Enabling scalable virtual organizations. International Journal of Supercomputer Applications (2001)
4. FIRE-MP User Guide, AVL List GmbH (1996)
5. The Globus Toolkit Project, `http://www.globus.org`
6. Cluster Resources Inc. Web Site, `http://www.clusterresources.com`
7. Karonis, N.T., Toonen, B., Foster, I.: MPICH-G2: A Grid-Enabled Implementation of the Message Passing Interface. J. Parallel and Distributed Computing 63(5), 551–563 (2003)
8. Snir, M., Otto, S., Huss-Lederman, S., Walker, D., Dongarra, J.: MPI: The Complete Reference. Massachusetts Institute of Technology (1996)
9. Patankar, S.V.: Numerical Heat Transfer and Fluid Flow. Hemisphere, Washington, DC (1980)
10. Nikseresht, A.H., Alishahi, M.M., Emdad, H.: Volume-of-Fluid Interface Tracking with Lagrangian Propagation for Incompressible Free Surface Flows. International Journal of Science and technology 12(2), 131–140 (2005)
11. Gueyffier, D., Li, J., Nadim, A., Scardovelli, R., Zaleski, S.: Volume-of-Fluid Interface Tracking with Smoothed Surface Stress Methods for Three-Dimensional Flows. J. of Computational Physics 152, 423–456 (1999)

Enhancing Accuracy of Source Localization in High Reverberation Environment with Microphone Array

Nima Yousefian, Mohsen Rahmani, and Ahmad Akbari

Department of Computer Engineering, Iran University of Science and Technology
ni_yousefiyan@comp.iust.ac.ir, {m_rahmani,akbari}@iust.ac.ir

Abstract. Array processing involves the use of multiple microphones to receive or transmit a signal carried by propagating waves. The classical method for source localization with microphone array is assumed to be Time Delay Estimation (TDE). Recently more attention is paid to another technique, interaural level difference (ILD) which estimates location of sound source by energy ratio that received from different microphones. In this paper we introduce a method based on combination of time delay likelihood and ILD that estimates source location in 2-D environment by two microphones. Results of experiments show that accuracy of this method is much better than previous methods proposed in this manner under normal conditions.

Keywords: Microphone Array, Source Localization, Time Delay Estimation.

1 Introduction

One of the most popular approaches for source localizations is algorithms that employ time-difference of arrival (TDOA) information [1]. Recently a distinct method which is based on interaural level differences (ILD) has been proposed. The signals received by the microphones not only differ in their relative time shift but also in their intensity level, with a microphone closer to sound source receiving a more powerful signal than that received by further microphone [2]. However in 2-D space all of above methods require at least three microphones to obtain exact position of sound source [3]. In this work we combine likelihood of TDOA and ILD-based approach, for estimating sound source location. Only two microphones are sufficient for estimation of source position with suitable accuracy.

In [3] a combination of TDOA and ILD has been proposed for 2-D space with dual-microphone system. This method gets position of source as intersection of two hyperbola and circle that calculated from TDOA and ILD separately. The problem is that method does not work with all situations for microphones and sound source positions and a specific relation between their distances should be held to obtain desirable results. For some situations where reverberation coefficients are above 0.8 this method gets source position out of region of interest. Also for generalization to 3-D space many changes in structure of algorithms in [3] are necessary but we may change our method to 3-D space much easier. Overall comparison between these two methods results will be discussed in simulation results section.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 803–806, 2008.

2 Model and Algorithm Description

We propose an algorithm based on TDOA and ILD that has three steps. In first and second steps localization error based on TDOA and ILD are computed respectively. In third step errors are combined to obtain an error that serves as localization criteria.

2.1 TDOA Maximum Likelihood Step

Consider a pair of microphones with spatial coordinates denoted by the 2-element vectors p_1 and p_2. For a signal source with known spatial location, q_s, the true TDOA to this sensor pair will be denoted by T ($\{p_1, p_2\}$, q_s) and calculated by difference of distance of each microphone to source divided by speed of sound in air [1]. In practice, The TDOA estimate (τ) is a corrupted version of the true TDOA and we have $\tau \neq$ T ($\{p_1, p_2\}$, q_s). Here TDOA is estimated by GCC-PHAT method. The maximum likelihood location estimate can be shown to be the position which minimizes the least squares error criterion which is defined as follow

$$Er_1(q) = \sum_{i=1}^{M} (\tau - T(\{p_{i1}, p_{i2}\}, q))^2 \tag{1}$$

Where M is the number of microphone pairs. As here only two microphones are present so M=1. We can rewrite (1) as follows

$$Er_1(q) = | \tau - T(\{p_1, p_2\}, q) |^2 \tag{2}$$

The estimated source is q with minimum error. Instead of finishing localization algorithm here by choosing best estimation, we choose N-best estimations and pass them to second step of algorithm which will be defined in next subsection. Number of points that are candidate for passing to second step will be defined later.

2.2 ILD-Based Step

According to the so-called inverse-square-law the energy received by *i-th* microphone can be modeled as

$$E_i = \int_0^W x_i^2(t)dt = \frac{1}{d_i^2}\int_0^W s^2(t)dt + \int_0^W n_i^2(t)dt \tag{3}$$

Where $n_i(t)$ is additive noise. d_i is the distance from source to *i-th* microphone and W is window length [2]. Energy received by *i-th* microphone can be obtained by integrating the square of the signal over this time interval. By considering this fact after some manipulation and neglecting energy of noise in relation to energy of clean signal we can conclude that energy received by each microphone has inverse proportion to square of distance of microphone to source. Let α be E_1 to E_2 ratio. Now the only procedure that we should follow is to estimate α by all of N-best points that earned from previous step. The estimation of α done by distances of microphones to assumed

sound source so $\hat{\alpha}(q) = \frac{d_2^2(q)}{d_1^2(q)}$ .The position that has $\hat{\alpha}$ closest to real α is best estimation and is considered as position of sound source. Actually this estimation is done by minimizing following error term.

$$Er_2(q) = |\alpha - \hat{\alpha}(q)| \tag{4}$$

2.3 Combining Two Methodologies

Experiments show that ILD based approaches are more sensitive to reverberation than TDOAs. So we decrease numbers of N-best points that are sent to the second step of algorithm as the reflection coefficients rise. By choosing this method for calculating N-best points we restrict areas that ILD estimates in reverberant environment. Then we combine errors earned from two mentioned steps, introduced in (2) and (4) to obtain final error. First, two error matrices should be normalized. By assuming β as reflection coefficient the final error value can be defined as

$$Er(q) = \beta * Er_1(q) + (1-\beta) * Er_2(q) \tag{5}$$

The point with minimum error is the final estimation of source location.

3 Simulation Results

We tested the proposed method in a 6m * 6m * 3m simulated room using Image method [5]. Figure 1 shows situation of components in an assuming test and results of algorithms at the end of each step.

Now we want to take a comparison between method proposed in this paper and method in [3]. Sound source is located one meter away from array center and along 30 and 45 degrees with respect to array normal direction. Errors of estimation sound source for two methods are in table 1. As it is clear from table 1 proposed method gives better results in reverberant situation. In some references another measure for calculating errors is considered. Mean angular error considers angel between estimated and real source points respected to array center as error rate [4]. We refer our method in table by A and method in [3] by B.

Table 1. Comparison of distance errors in cm (left) and mean angular error in degrees (right) in source estimation between proposed method (A) and method in [3] (B)

Reflection Coefficient	θ=30 A	θ=30 B	θ=45 A	θ=45 B	θ=30 A	θ=30 B	θ=45 A	θ=45 B
0.6	1.6	3.2	10.7	13.7	0.0184	0.022	0.744	0.487
0.7	4.1	7.3	11.2	13.3	0.465	0.467	0.745	0.476
0.8	11.5	11.2	12.1	18.7	0.173	0.113	0.002	0.182
0.9	6.5	9.2	18.3	25.8	0.784	0.878	0.002	0.137

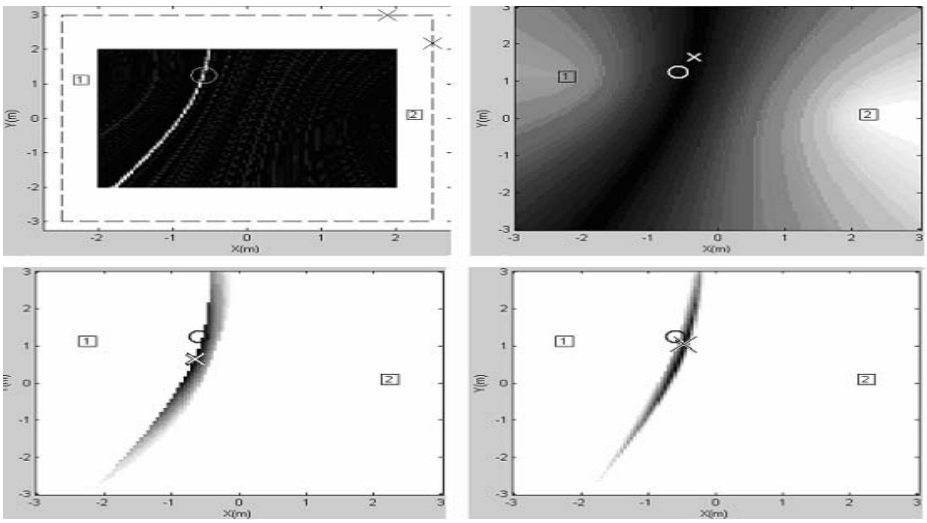


Fig. 1. (Top Left): Sound source position is in (O), microphones are in squares and two (X) denote noise sources. (Top Right): Errors and sound source estimation at the end of the first step of algorithm. Estimated point is shown by (X). (Bottom Left): Estimation of sound source at the end of the second step of algorithm. Black areas denote N-best points. (Bottom Right): Final estimation of sound source position. Great improvement in the source estimation is clear.

4 Conclusion

We introduced three step algorithms that combine two well known approaches TDOA and ILD, in source localization. Results show that it improves accuracy of localization estimation. Despite recently proposed method as in [3] our technique works well for reverberant environments too.

References

1. DiBiase, J., Silverman, H., Brandstein, M.: Robust Localization in Reverberant Rooms, Microphone Arrays: Signal Processing Techniques and Applications. Springer, Heidelberg (2001)
2. Birchfield, S.T., Gangishetty, R.: Acoustic localization by interaural level difference. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 4, pp. 1109–1112 (2005)
3. Weiwei, C., Zhigang, C., Jianqiang, W.: Dual-Microphone Source Location Method in 2-D Space. In: IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 4, pp. 845–848 (2006)
4. Jean-Marc, V., François, M., Jean, R., Dominic, L.: Robust Sound Source Localization Using a Microphone Array on a Mobile Robot. In: International Conference on Intelligent Robots and Systems, vol. 2, pp. 1228–1233 (2003)
5. Allen, J.B., Berkley, D.A.: Image method for efficiently simulating small-room acoustics. J. Acoust. Soc. Am. 65(4), 943–950 (1979)

Class Dependent LDA Optimization Using Genetic Algorithm for Robust MFCC Extraction

Houman Abbasian, Babak Nasersharif, and Ahmad Akbari

Research center for Information Technology, Computer Engineering Department,
Iran University of Science and Technology
hooman_abbasian@comp.iust.ac.ir,
{nasser_s,akbari}@iust.ac.ir

Abstract. Linear Discrimination analysis (LDA) finds transformations that maximizes the between-class scatter and minimizes within-class scatter. In this paper, we propose a method to use class-dependent LDA for speech recognition and MFCC extraction. To this end, we first use logarithm of clean speech Mel filter bank energies (LMFE) of each class then we obtain class-dependent LDA transformation matrix using multidimensional genetic algorithm (MGA) and use this matrix in place of DCT in MFCC feature extraction. The experimental results show that proposed speech recognition and optimization methods using class-dependent LDA, achieves to a significant isolated word recognition rate on Aurora2 database.

Keywords: Linear Discrimination analysis, Class-dependent, MFCC, Multi-dimensional genetic algorithm, Speech recognition.

1 Introduction

Feature extraction is a crucial step of speech recognition process. Among speech feature extraction methods, MFCC features are more commonly used for speech recognition. The MFCC features are obtained by applying discrete cosine transform (DCT) to logarithm of Mel filter bank energies (LMFE) [1]. There are several techniques that improve MFCC features from different points of view. DCT is a non-adaptive procedure that projects LMFE in the direction of global variance which achieves only partially decorrelate speech features. (LDA) [2] is one of the approaches which replaces (DCT) in MFCC in order to improve MFCC.LDA computes a transformation (projection) by minimizing the within class scatter and maximizing the between-class scatter simultaneously. One limitation of LDA is that it ceases to work well when classes are not linearly separable. Several generalizations of LDA have been proposed to address this limitation. Two examples of such approaches are: nonlinear LDA (NLDA) [2] and kernel LDA (KLDA)[3]. We propose to optimize class-dependent LDA (CD-LDA) transformation using multi dimensional genetic algorithm in order to achieve more discrimination among classes. We name this method as GCD-LDA. We apply the optimized CD-LDA transformation in place of DCT to LMFE. Using this transformation, LMFE features will be more uncorrelated and so their covariance matrix will be more diagonal. This results in better HMM

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 807–810, 2008.

training in each class and more discrimination among classes. In section 2, we describe the main idea of this method. Section 3, briefly explains multidimensional genetic algorithm for optimization of class-dependent LDA transformation matrix. Finally, section 4 includes our experiments and results.

2 LDA Optimization Using Multidimensional Genetic Algorithm

In this section, we propose to optimize LDA transformation matrix in order to make speech classes more separated from each other. For this purpose, we optimize CD-LDA using multidimensional genetic algorithm. We named our method as GCD-LDA. When we apply the optimized LDA transformation on LMFE features, speech classes will be more separated from each other. These transformed features, are less sensitive to noise due to optimized class-specific transformation. Thus, recognition rate is improved in presence of noise.

If we assume to have K speech signal in a class i and for each speech signal, we have $L \times M_i$ LMFE matrix in which L is the number of Mel Filters and M_i is the total number of frames in i-th signal. If *c* shows the number of classes of speech units, we need *c* CD-LDA transformation matrices of size $L \times L$. We use Multidimensional Genetic Algorithm (MGA) in order to obtain these *c* transformation matrices, simultaneously. For this purpose, we define each Individual as a $L \times L \times c$ matrix. Then, each individual is considered as *c* transformation matrices of size $L \times L$. Using genetic algorithm, we can find best individual and so best transformation matrix which maximizes the following criteria:

$$J(W) = \sum_{i=1}^{c} \frac{\left| W_{i,CD}^{T} S_B W_{i,CD} \right|}{\left| W_{i,CD}^{T} S_{W,CD}^{i} W_{i,CD} \right|} \tag{1}$$

In this criteria $S_B, S_{W,CD}^{i}$ are defined as follows:

$$\begin{aligned} S_{W,CD}^{i} &= \sum_{j=1}^{M_i} (x_i(j) - \mu_i)(x_i(j) - \mu_i)^T \\ S_B &= \sum_{i=1}^{c} (\mu_i - \mu)(\mu_i - \mu)^T \end{aligned} \tag{2}$$

Where $x_i(j)$ shows j-th speech signal of class i. μ_i is the mean of class i , μ represents the mean of all classes, c is number of all classes and M_i is number of all frames (samples) in class i.

3 Multidimensional Genetic Algorithm Parameters

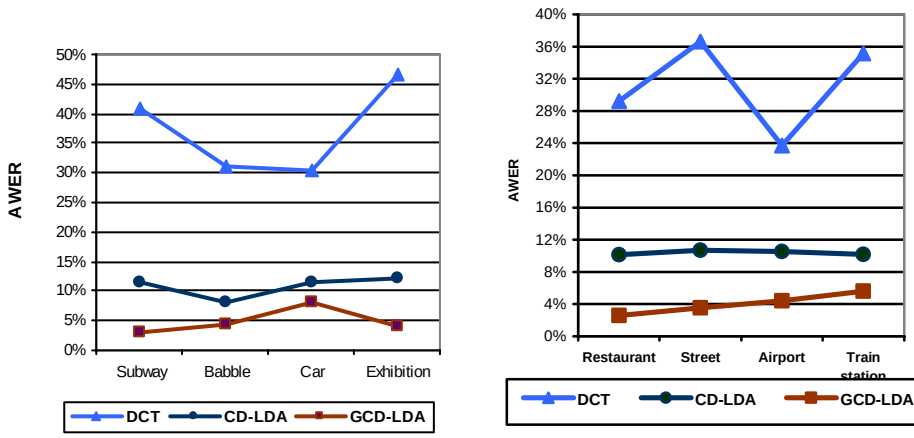
The principles of Genetic Algorithms consist of maintaining and manipulating a population of solutions and implementing a 'survival of the fittest' strategy in their search for better solutions [4].

For any GA, a chromosome representation is needed to describe each individual in the population. The representation scheme determines how the problem is structured in the GA and also determines the genetic operators that are used. In response to huge computational demands, representation system can be designed in a multi dimensions manner. We name such GA as Multidimensional GA (MGA). In such way, each individual can be represented by a cube. It is obvious that we should define appropriate mutation and crossover for this new representation. In our case, this is a multi-dimensional real-valued representation .In this paper, we want to obtain all LDA transformation matrices corresponding to each class. We use multi- dimensional real-valued individuals. Here our representation is three dimensional matrix, $L \times L \times c$. Where L is the number of Mel filter banks and c is the number of classes of speech units. In this work, we want to generate c functions simultaneously. So, each individual must contain c different mapping in size $L \times L$ which L is the number of Mel filter banks. After generating initial population, fitness of each individual should be computed. Fitness is a measure which represents how well an individual is. For selecting appropriate fitness function, we should reconsider our algorithm objective. As mentioned before, our main objective function is to generate transform matrices for each class, such that maximize LDA criterion. Therefore, we can calculate our fitness function as in equation (1).In this paper, we use tournament selection [5] with tournament size 5 to select best members of population based on their fitness.

As mentioned in above, we need to propose new mutation and crossover operators for our method. To do this, three stage crossover is applied. Each stage of this crossover operation is like general crossover. But here, general crossover is applied in three dimensions, during each step. In this work, mutation operation consists of randomly generating numbers in the interval [-1,1] and then adding these numbers to the original genes of each individual. The result of this addition stay in the interval [-1,1].After using genetic operators, we should decide which members of old population should be replaced by new individuals to generate new population. For this purpose, we use elitism method. In this method, a part of old population (Generation Gap) is copied unchanged to the next population [5]. In this paper, 10% of fittest old population are copied unchanged to the next population and remaining members of old population are replaced by new individuals. This procedure repeats until the maximum number of generations is reached. Here in this work our maximum generation is 100.

4 Results and Discussion

We report our results on Aurora 2 database for isolated word recognition. Only clean data are used for HMM training. Our recognizer is CDHMM with 16 states and 3 Gaussian mixtures per state. There are 8 types of noises in the 3 test sets: sets A, B and C. Our feature vector in all cases contains 12 projected LMFE (MFCC in case of DCT) and 12 delta-coefficients and so its length is 24. In GCD-LDA and CD-LDA, we use only clean training set. In this way, we determine our transformation matrix in a class-dependent manner based on CD-LDA and GCD-LDA.



(a) Set A (b) Set B

Fig. 1. Average word error rate over all SNR values separated for different noise types and three test sets (A,B)

Fig. 1 shows average word errors rate (AWER) over all SNR values (20, 15, 10, 5,0) which are separated for different types of noise and two test sets. Words DCT, CD-LDA and GCD-LDA show the type of transformation matrix applied to LMFE. So, DCT indicates the conventional MFCC features. CD-LDA and GCD-LDA show cases that DCT matrix is replaced by class-dependent LDA and MGA based optimized class dependent LDA transformation matrix in MFCC features extraction. As can be seen form the figure, for both 2 test sets, CD-LDA has lower AWER than DCT. In addition, Fig. 1 shows that optimized CD-LDA using MGA (GCD-LDA) has lower AWER than CD-LDA for MFCC extraction. This is noticeable because we don't use any information of noise in determining GCD-LDA and also CD-LDA transformation matrices. This can be due to that we can more discriminate classes using class-specific transformation. Thus, recognition results are less affected in presence of noise.

References

1. Nasersharif, B., Akbari, A.: A Framework for Robust MFCC Feature Extraction Using SNR-Dependent Compression of Enhanced Mel Filter Bank Energies. In: International Conference on Spoken Language Processing (ICSLP), pp. 33–36 (2006)
2. Somervuo, P.: Experiments with linear and nonlinear feature transformations in HMM based phone recognition. In: IEEE Int. Conf. on Acoustics, Speech and Signal processing, vol. 1, pp. 52–55 (2003)
3. Scholkopf, B., Smola, A.J., Muller, K.R.: Nonlinear component analysis as kernel eigenvalue problem. Neural computation 10, 1299–1319 (1998)
4. Gabris, B., Ruta, D.: Genetic algorithms in classifier fusion. Applied soft computing 6, 337–347 (2006)
5. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading (1989)

TACtic- A Multi Behavioral Agent for Trading Agent Competition

Hassan Khosravi[1], Mohammad E. Shiri[1], Hamid Khosravi[2], Ehsan Iranmanesh[1], and Alireza Davoodi[1]

[1] Department of Computer Science, Faculty of Mathematics and Computer Science
Amirkabir University of Technology, Tehran, Iran
hkhosrav@cs.sfu.ca Shiri@aut.ac.ir, iranmanesh@aut.ac.ir, davoodi@aut.ac.ir
[2] International Center for Science & High Technology and Environmental Science
Kerman, Iran
khosravi@icst.ac.ir

Abstract. Software agents are increasingly being used to represent humans in online auctions. Such agents have the advantages of being able to systematically monitor a wide variety of auctions and then make rapid decisions about what bids to place in what auctions. They can do this continuously and repetitively without losing concentration. To provide a means of evaluating and comparing (benchmarking) research methods in this area the trading agent competition (TAC) was established. This paper describes the design, of TACtic. Our agent uses multi behavioral techniques at the heart of its decision making to make bidding decisions in the face of uncertainty, to make predictions about the likely outcomes of auctions, and to alter the agent's bidding strategy in response to the prevailing market conditions.

Keywords: Game Theory, Multi behavior, intelligent agents, online auctions, trading agent competition (TAC).

1 Introduction

AGENT-MEDIATED electronic commerce involves software agents acting on behalf of some or all the parties in e-commerce transactions [4], [6]. Here, a software agent is viewed as an encapsulated computer system, situated in some environment that is capable of flexible autonomous action in that environment in order to meet its design objectives [7]. The rationale for introducing such agents in e-commerce scenarios is to offer faster, cheaper, more convenient and more agile ways for both customers and suppliers to trade. As the agents represent distinct stakeholders and organizations, the *de facto* way in which they interact is through some form of *negotiation*. In human negotiations, two or more parties typically bargain with one another to determine the price or other transaction terms [2].

Given the potential and the importance of using agents in online auction settings, there has been considerable research endeavor in developing bidding strategies for different types of agents in different types of auctions (see Section V for more

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 811–815, 2008.

details). Therefore, in order to develop a means of comparing and evaluating this work, it was decided to establish an International Trading Agent Competition (TAC) (similar in spirit to other initiatives such as RoboCup,[3] RoboCupRescue[4] and the Planning Competition[5]). In this competition, software agents compete against one another in 28 simultaneous auctions in order to procure travel packages (flights, hotels and entertainment) for a number of customers (see Section II for more details of the roles).

2 Introducing Trading Agent Competition

In each TAC trading game, there are eight software agents that compete against each other in a variety of auctions to assemble travel packages for their individual customers according to their preferences for the trip. A valid travel package for an individual customer consists of: i) a round trip flight during a five-day period (between TAC-town and Tampa), and ii) a stay at the same hotel for every night between their arrival and departure dates. Moreover, arranging appropriate entertainment events during the trip increases the utility for the customers. The objective of each agent is to maximize the total satisfaction of its eight customers (i.e., the sum of the customers' utilities).

3 TACtic

Our design is quite different from other agents. TACtic has inherited its characteristics from 3 successful agents. RoxyBot, Attac-2000 and SouthamtonTAC are the agents that we have used. These agents are advantageous on different situations. TACtic uses a Probabilistic learning automaton with 4 different behaviors to achieve the best outcome. These behaviors are as follows and will be discussed in more details later on. In our implementation we use riskful, economical, computational, and prudent behavior.

TACtic uses Attac-2000's rules on risk averse agents and SouthamtonTAC's rules on airplane ticket division to split the competition into 5 different environments as shown. Table1 shows these environments.

Table 1. Different environments of tac that tactic uses

Plane Types / Number of Risk agents	Cat0	Cat1	Cat2	Cat3
1,2	4	3	2	1
3,4	5	4	3	2
5,6,7	6	5	4	3

TACtic uses the 6 mentioned environments with the automaton to learn which behavior is most suitable for different environments. The extended version of the paper fully discuses the states of the automaton.

This automaton tries learning what behavior works better for what environment. It increases and decreases the probabilities to converge to the right values. We believe

that the competition's environment is very dissimilar and it needs different behaviors to maximize its utility at different places. We will try to explain these behaviors separately.

Riskful behavior: The riskful behavior is actually pretty common in TAC. For buying airplane tickets in this behavior, we ask the allocator about our needs at the beginning of the game. Most of the plane tickets needed are bought without considering their price. After receiving the requirements from the allocator, the agents bids high enough to get the plane tickets.

Since the agent has bought most of the plane tickets required, it doesn't have much flexibility on hotel reservation. The agent bids high knowing that it needs to get a hold of these rooms. In this behavior the agent works out the maximum price in which the hotel is worth getting and uses that price to bid in the auctions. On entertainment goods, the agent knows exactly when each of its customers are arriving and leaving. This will make it much easier to decide on what to buy. The agent works like Attac-2000 on goods that he has but doesn't need them.

Economical behavior: This behavior is completely new in TAC. As far as our research goes, we haven't seen any agents that use this method. For buying airplane tickets, The agent starts looking through the airplane auctions. In this behavior buying is not done through a list, but everything that seems to have reasonable price is bought. Here the agent doesn't consider the problem of allocating the good, but fully concentrates on the prices in the auctions. Any plane ticket cheaper than $320 is bough immediately. After the fourth minute the agent uses the allocator before bidding in auctions.

For bidding in hotel auctions the agent needs to be active. The agent tries bidding on the 15 cheapest rooms which are atleast from 4 different auctions. For doing so the agent should have a sorted array of all of the auction prices at all times which is a simple algorithm but require a lot of time. In entertainment goods the same algorithm is used. The agent tries bidding on the cheapest goods of the market and sells its most expensive tickets. This procedure is carried out for 4 minutes. After that the agent uses the allocator to decide on its biddings.

Computational behavior: This behavior is based on prediction mostly. For bidding in plane auctions, the agent uses SouthamtonTAC's method which is actually very efficient [9]. For hotel auctions, Roxybots method of allocation [5] is used. For finding the maximum benefit that the room can have, we calculate our utility using Roxybot's allocation method once with and once without the room. The difference between these two numbers will give us the maximum benefit that the reservation of the room has for us.

In entertainment auctions, first each of the 12 tickets that we have, are valued the same way as hotel rooms values were found. Since 24 allocations should be done Roxybot's algorithm takes too long to use. We use the 100 arrangements [15] prepared by Attac-2000 which is reasonably fast. The same algorithm is used for buying goods. We calculate our utility with and without the ticket which will give us the value of the ticket.

Prudent behavior: This behavior is based on not committing to any plans for the future. The agent in this behavior buys a few airplane tickets at the beginning of

the game. These tickets will be bought from the two most popular auctions for the agent. Since the agent wants many tickets from these auctions, buying half of them at the beginning doesn't really bring much commitment.For hotel auctions, since the agent doesn't have any commitment towards any plan, it can choose from cheaper hotel rooms. Actually the agent starts making commitment by reserving low-priced hotel rooms.

At the beginning of the game the agent only buys goods for nights which it thinks more than 5 customers are on holiday. This way it doesn't face great risks in committing. The rest of the tickets are bought as game goes by and plans are made. Attac-2000 method for finding the right price is used.

The agent also has architecture and a way of allocating goods which due to space limitations will not be discussed here and will be given a longer version of the paper.

5 Conclusion and Future Works

This paper presents a successful application of multi behavioral acts in agent-mediated electronic commerce. It details the design of TACtic an agent that employs a range of techniques at its core. Specifically, it uses the advantages of different other successful agents to determine the type of environment it is situated in and then uses an adaptive bidding strategy to change its strategy depending on this assessment. TACtic has been shown to be successful across a wide range of TAC environments. Naturally the strategies that have been employed are tailored to the specific auction context of the competition. Nevertheless, we believe that the TAC domain exhibits a number of characteristics that are common to many real-world online trading environments. These attributes include a time constrained environment, network latency, unpredictable opponents, multiple heterogeneous auction types, and the need to purchase inter-related goods. Given this, we believe that a number of technologies and insights from our work are applicable in a broader agent-mediated e-commerce context and our future work aims to exploit these.

References

[1] Fisher, R., Ury, W.: Getting to Yes: Negotiating an Agreement Without Giving. Random House, New York (1981)
[2] Friedman, D., Rust, J.: The Double Auction Market: Theories and Evidence. Addison-Wesley, Reading (1992)
[3] Guttman, R.H., Moukas, A.G., Maes, P.: Agent-mediated electronic commerce: A survey. Knowledge Eng. Rev. 13(2), 147–159 (1998)
[4] Greenwald, A., Bovan, J.: Bidding Algorithm for simultaneous auctions (2002)
[5] He, M., Jennings, N.R., Leung, H.F.: On agent-mediated electronic commerce. IEEE Trans. Knowledge Data Eng. 15, 985–1003 (2003)
[6] Jennings, N.R.: An agent-based approach for building complex software systems. Commun. ACM 44(4), 35–41 (2001)
[7] He, M., Jennings, N.R.: SouthamptonTAC: Designing a successful trading agent. In: van Harmelen (ed.) Proc. 15th Eur. Conf. Artificial Intelligence, Amsterdam, The Netherlands, pp. 8–12 (2002)

[8] Khosravi, H., Shiri, M.: design of an intelligent agent for multi agent environments using game theory. Master Thesis in Amirkabir University (May 2007)
[9] Preist, C., Byde, A., Bartolini, C.: Economic dynamics of agents in multiple autions. In: Proc. 5th Int. Conf. Autonomous Agents, Montreal, QC, Canada, pp. 545–551 (2001)
[10] Stone, P., Littman, M.L., Singh, S., Kearns, M.: Attac-2000: An adaptive autonomous bidding agent. J. Artif. Intell. Res. 15, 189–206 (2001)
[11] Wurman, P.R., Wellman, M., Walsh, W.: A parameterization of the auction design space. Games Econ. Behavior 35, 304–338

Software Reliability Prediction Based on a Formal Requirements Specification

Hooshmand Alipour[1] and Ayaz Isazadeh[2]

[1] Islamic Azad University-Pars abad Moghan, Pars abad Moghan, Iran
halipour@iaupmogan.ac.ir
[2] Department of Computer Science, Tabriz University, Tabriz, Iran
isazadeh@tabrizu.ac.ir

Abstract. Software reliability models are mostly used at the test phase; there are only a few models that are employed at early phase of software development. Early prediction, however, is very important for better prognosis and management of risks. In this paper we propose an approach for early software reliability prediction, based on software behavioral requirements. The major difference between our approach and those of others is the fact that we use a formal method, called Viewcharts, to specify the behavior of software systems.

Keywords: software reliability models, formal method, requirement phase.

1 Introduction

Reliability is one of the major factors of software quality and is defined as the "*probability of failure-free software operation for a specified period of time in a specified environment*" [2]. Based on this definition, if Q(t) is the probability of failures during time (0,t], then software reliability is measured as follows:

$$R(t)=1-Q(t) \quad (1)$$

Using this definition, for an accurate software reliability measurement, the behavior of software (at the failure occurrence viewpoint) must be studied. In large-scale and complex systems, validation and verification of software requirements are important issues. Statistically speaking, 80% of all the defects in software systems are inserted during the requirements phase [1]; any modification and change in user requirements may cause manipulation of up to 95% of the system at the maintenance phase. It is necessary, therefore, to asses and improve the reliability of software systems at the requirement phase. For early prediction of software reliability we need to specify the system using some formal description technique and study the probability of its failure states [5].

In this paper we use a formal method, called Viewcharts [2] for software system behavioral description. Viewcharts is designed for specifying the behavior of a system at requirement phases and, therefore, it can be used as a starting point for early software reliability prediction. We describe this method with an example, demonstrating its applicability for early prediction of software reliability. First, we make some minor

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 816–820, 2008.

modifications to Viewcharts, making it suitable for our purpose. These modifications contain entering Markov chain concept into Viewcharts method and some states to each view in viewchart specification of system. Then we show the way in which software reliability can be measured, using Viewcharts, at the requirements specification phase of software systems.

2 Related Work

There are just a few attempts, addressing the concept of *early* software reliability prediction (specially in requirement phase). Gaffney and Davis [9] developed the phase-based model. It makes use of error statistics obtained during the technical review of requirements, design and the implementation to predict the reliability during test and operation. This model expects that fault densities will be expressed in terms of the number of faults per thousand lines of source code (KSLOC), which means that faults found during the requirements analysis and software design will have to be normalized by the code size estimates. Another well known work in this field initiated by RADC [10]. To early prediction of software reliability, the researchers applied some requirements and design representation metrics to obtain the initial fault density. On the other hand there are some researches which uses the output of design (directly or indirectly) to reliability prediction; these are like to our model. Both of architecture-based [6], [7], [12], [13] and structure-based [14], [15] models are as mentioned models. Architecture-based and structure-based models are similar. These models seek to assess to behavior of software application taking into consideration the behavior of its parts (components) and the interaction between its parts. Most of these models are dependent to the real failure data obtained in the test phase. Since our model assesses the software reliability by taking into consideration the software behavior, it is like to the architecture and structure based models. But our model tries to predict the software reliability in the requirement phase. We believe that our model is more accurate than the similar models, because it studies the software behavior based on a formal method requirement specification language.

A Behavioral view of a software system is the behavior of the system observable from a specific point of view. The Viewcharts notation is based on Statecharts [8]. Statecharts, however, has no concept of behavioral views. Viewcharts extends Statecharts to include views and their compositions. Three types of views compositions are in the Viewcharts: SEPARATE, AND and OR compositions. Fig.1, Fig.2 and Fig.3 illustrate these compositions.

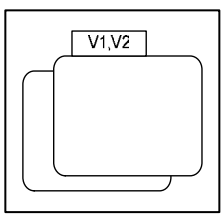


Fig. 1. SEPARATE composition

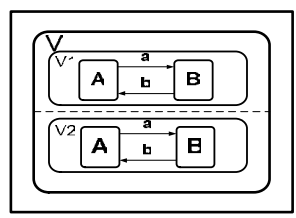


Fig. 2. AND composition

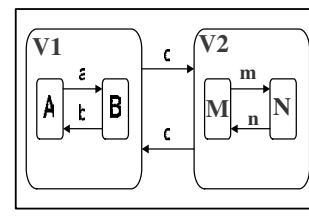


Fig. 3. OR composition

In a **SEPARATE** composition of views, all the views are active; no transition between the views is allowed, the scopes of all the elements are unaffected; and any subview or state in one view is hidden from (i.e., cannot be referenced by) the other views. In an **AND** composition of views, all the views are active; the scopes of all the elements owned by each view are extended to the other views. The **OR** and SEPARATE compositions are similar, except that in an OR composition, only one view can be active and there can be transitions between the views. For more details about Viewcharts refer to [1].

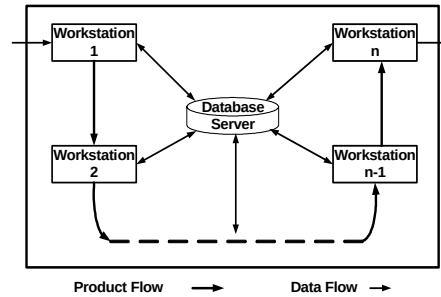


Fig. 4. Information representation of MSC

We now present the Viewcharts specification of a Manufacturing Control System (MCS) and then measure its reliability. It consists of a number of workstations, where each workstation performs a certain process on the product. Central to the system is a database server (DBS) which maintains and supplies the information requirements of the workstations. At the beginning of the manufacturing line, the first workstation associates each product with a unique identification number/string pid, which must be communicated to DBS to create a record for the corresponding product. When a product arrives at a workstations, the pid is scanned and communicated to DBS which, in turn, informs the workstation of the process that must be performed on the product. The workstation then proceeds with the process and when it is completed, informs DBS to update the product record. We use a single but compound variable db to represent the MCS database. Fig.4 shows an informal diagram of MCS. Fig.5 illustrates the viewchart specification of system; it consists of two anded views: WS, which describes the behavior of the workstation, and DBS, which describes the workstation's view of DBS.

3 Assessment of Software Reliability Using Viewcharts

In this section we present our approach to software reliability prediction using formal software behavioral requirements specification represented by Viewcharts. But, first we need to make a modification to Viewcharts.

3.1 Modified Viewcharts

Basically, Viewcharts is a model for description of system in early phases of development. For using its benefits for showing states of system and then prediction of software reliability we need to add the concepts of Markov chain to it. Therefore, in each view instead of Statecharts we use Markov chain. For applying these modifications, we first draw Viewcharts specification of system and then add rate of each transition on its arc. Fig.6 illustrates the modified Viewcharts for MCS system.

In this model, we should anticipate system (or parts of system) failure states. For each view we should fined events which if they occur then the system configuration

will be false (fault will occur). According to these events some states is specified to each view. So if a faulty event is occurred in the specific view then the mentioned states will be active. For finding rate of system states transition, we can study similar systems or older version of current system. The best way to obtain the rate of transitions is developing the software for simulation of this model. We designed the software that makes possible to a requirement engineer to specify and simulate a software system in the requirement phase and implementation of this system is our future work. According to each use case the Viewchart simulator selects random subset of system inputs or triggers some of the external events. After selecting input(s) or triggering external event(s) the simulator traces the affect of them and finally when the simulation of specific input finished, then the resulting configuration of the system in viewchart specification is recorded. After finishing the simulation of all inputs or events affect, the transition rates between all states will be obtained.

3.2 Reliability of System

According to description in previous section, suppose Figure 6 is a modified Viewcharts from a system. In this figure unknown and down states are two failure states that are considered for the WS1 and DBS views. If we can obtain probability of failure states then reliability of system is calculated as follow:

$$R = 1 - \left(prob(UNKNOWN) \cup prob(DOWN) \right) \tag{2}$$

For computation of system failure states probability we must use Markov chain conservation low. If P_1= prob(SERYALIZING), P_2=prob(WAIT) and P_3=prob(UNKNOWN) then For WS view this low is as follows:

$$\begin{cases} \lambda_1 p_1 = \lambda_2 p_2 + \lambda_4 p_3 \\ (\lambda_2 + \lambda_3) p_2 = \lambda_1 p_1 \\ \lambda_4 p_3 = \lambda_3 p_2 \end{cases} \tag{3}$$

After solving of above equation, p_3 is obtained.

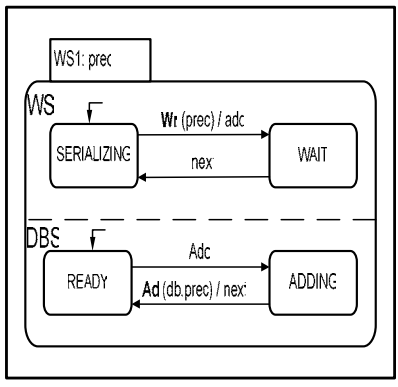


Fig. 5. A viewchart for the Serialization Workstation

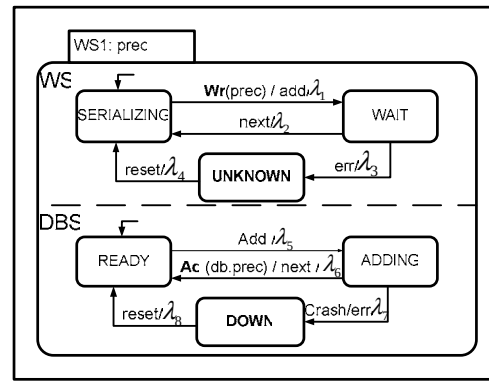


Fig. 6. Modified Viewcharts

4 Conclusion

We have introduced a new approach for representing the behavior of software systems and using it for early software reliability prediction. Our approach combined two methods, Markov chain and Viewcharts, and showed the states of system. The Viewcharts method has the ability to specify heterogeneous and complex systems easily. On the other hand, for adding the rate of system's transition, from a state to another one, we have used Markov chain. We also predicated some states for showing system failures and added these states to our model. Consequently, we have provided an early prediction of the software system reliability as the union of failure states probability. The main characteristics of our paper are: applicability at the requirement phase, independency from real failure data, flexibility with respect to requirement changes, formality and clarity with respect to reasoning and deduction.

References

1. Isazadeh, A., Lamb, D.A., Shepard, T.: Behavioral views for software requirements engineering. Requirements Engineering Journal 4(1), 19–37 (1999)
2. Standard Glossary of Software Engineering Terminology, STD-729-1991, 1991; ANSI/IEEE
3. Yin, M.-L., Hyde, C.L., James, L.E.: A Petri-Net approach for earl-Stage system-Level software reliability estimation. In: Proceedings Annual reliability and maintainability Symposium. IEEE, Los Alamitos (2000)
4. Wang, W.-L., Pan, D., Chen, M.-H.: Architecture-based software reliability modeling. The Journal of Systems and Software 79, 132–146 (2006)
5. Wang, W.-L., Chen, M.-H.: Heterogeneous Software Reliability Modeling. In: Proceedings of the 13th International Symposium on Software Reliability Engineering (ISSRE 2002). IEEE, Los Alamitos (2002)
6. Harel, D.: Statecharts: A visual formalism for complex systems. Science of Computer Programming 8, 231–274 (1987)
7. Gaffney Jr., J.E., Davis, C.F.: An Approach to Estimating Software Errors and Availability., SPC-TR-88–007, version 1.0 (March 1988a); Proceedings of the 11th Minnowbrook Workshop on Software Reliability (July 1988)
8. McCall. J.: Rome Laboratory (RL), Methodology for Software Reliability Prediction and Assessment. Technical Report RL-TR-92-52, vol. 1, 2 (1992)
9. Gokhale, S.S., Wong, W.E., Trivedi, K.S., Horgan, J.R.: An Analytical Approach to Architecture-based Software Reliability Prediction. In: Proceedings of IEEE International Computer Performance and Dependability Symposium (IPDS), Durham, North Carolina (September 1998)
10. Wang, W.-L., Scannell, D.: An Architecture- Based Software Reliability Modeling Tool and Its Support for Teaching. In: Proceedings of the 35th ASEE/IEEE Frontiers in Education Conference (October 2005)
11. Gokhale, S.S., Lyu, M.R.: A simulation approach to structure- Based software reliability analysis. IEEE Trans. Software Eng. 31(8), 643–656 (2005)
12. Gokhale, S., Trivedi, K.: Structure-based software reliability prediction. In: Proceedings of the Fifth International Conference on Advanced Computing (ADCOMP 1997), pp. 447–452 (1997)

ID-Based Blind Signature and Proxy Blind Signature without Trusted PKG*

Yihua Yu, Shihui Zheng, and Yixian Yang

Information Security Center, State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, P.R. China
yuyihua2004@yahoo.com.cn, shihuizh@gmail.com, yxyang@bupt.edu.cn

Abstract. Private key escrow is an inherent disadvantage for ID-based cryptosystem, i.e., the PKG knows each signer's private key and can forge the signature of any signer. Blind signature plays a central role in electronic cash system. Private key escrow is more severe in electronic cash system since money is directly involved. To avoid the key escrow problem, we propose an ID-based blind signature and proxy blind signature without trusted PKG. If the dishonest PKG impersonates an honest signer to sign a document, the signer can provide a proof to convince that the PKG is dishonest.

1 Introduction

Blind signature plays a central role in electronic cash system where the bank plays as the signer and the customer plays as the user. A signature issued by the bank is equivalent to an electronic coin. Proxy blind signature satisfies the security properties of both the blind signature and the proxy signature. In electronic cash system, the user makes the bank blindly sign a coin using blind signature schemes. When a user goes through a valid branch to withdraw a coin, he needs the branch to make proxy blind signature on behalf of the signee bank.

In 1984, Shamir [5] proposed the concept of ID-based cryptosystem to simplify key management procedures of certificate-based public key systems. However, there is an inherent disadvantage in ID-based cryptosystem: private key escrow. Since the PKG knows each signer's private key, he can forge the signature of any signer. In electronic cash system, it is more severe since money is directly involved. Recently, researchers have been trying to solve the key escrow problem [4].

In this paper, we propose an ID-based blind signature and an ID-based proxy blind signature without trusted PKG. A PKG, which is not treated as an honest party, still generates private key of the signer and some secret information chosen by the signer is embedded in the private key. If the PKG impersonates a signer to sign a document, the signer can prove with the secret information that the PKG is dishonest.

* This work is supported by Sony (China) research laboratory. This work is supported by National Natural Science Foundation of China (No. 90604022, 60673098), and National Basic Research Program of China (973 Program) (No. 2007CB310704).

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 821–824, 2008.

2 ID-Based Blind Signature without Trusted PKG

Our ID-based blind signature scheme without trusted PKG is based on Zhang-Kim's ID-based blind signature scheme under the trusted PKG [6] and Boldyreva's blind signature scheme [1]. In the following, we use the notation $a \in_R A$ to mean that a is randomly chosen from A.

Let G_1 be a GDH group generated by P, whose order is a prime q. The bilinear pairing is given as $e: G_1 \times G_1 \to G_2$. Define three cryptographic hash functions $H_1: \{0,1\}^* \times G_1 \to G_1$, $H_2: \{0,1\}^* \times G_1 \to Z_q$ and $H_3: \{0,1\}^* \to G_1$.

Setup: PKG chooses $s \in_R Z_q^*$ and sets $P_{pub} = sP$. PKG publishes system parameters $param = \{G_1, G_2, e, q, P, P_{pub}, H_1, H_2, H_3\}$ and keeps s as the master-key.

Extract: A signer submits his identity ID and authenticates himself to PKG. The signer then chooses $r \in_R Z_q^*$ as his partial private key and sends rP to PKG. PKG computes $Q_{ID} = H_1(ID, rP)$ and $S_{ID} = sQ_{ID}$. The signer's private key is (S_{ID}, r).

Sign: Suppose that m is the message to be signed.

- The signer chooses $t \in_R Z_q^*$, computes $U = tH_1(ID, rP)$, and sends U to the user as a commitment. He also sends rP to the user.
- (**Blinding**) The user chooses $\alpha, \beta \in_R Z_q^*$ as blinding factors. He computes $R = \alpha U + \alpha\beta H_1(ID, rP)$ and $h = \alpha^{-1} H_2(m, R) + \beta$, sends h to the signer. The user chooses $\xi \in_R Z_q^*$, computes $M = \xi^{-1} H_3(m)$ and sends M to the signer.
- (**Signing**) The signer sends back V and B, where $V = (t + h)S_{ID}$ and $B = rM$.
- (**Unblinding**) The user computes $S = \alpha V$, $C = \xi B$. He outputs (m, R, S, C, rP).

Then (R, S, C, rP) is the blind signature of the message m.

Verify: Compute $Q_{ID} = H_1(ID, rP)$ and accept the signature if and only if

$$e(S, P) = e(R + H_2(m, R)Q_{ID}, P_{pub}),$$

$$e(C, P) = e(H_3(m), rP).$$

Suppose PKG wants to impersonate a signer whose identity is ID. He can do as follows: PKG chooses a random $r' \in_R Z_q^*$ and computes $S'_{ID} = sH_1(ID, r'P)$. He then performs the above described signing protocol on a message m to produce a signature $(R', S', C', r'P)$, which satisfies the verify protocol and is a "valid" signature.

If the PKG misattributes the signature to frame the signer, he signer can provide a proof to convince that the PKG is dishonest. The signer sends rP to the arbiter, and then provides a "knowledge proof" that he knows $S_{ID} = sH_1(ID, rP)$: the arbiter chooses $\alpha \in_R Z_q^*$ and sends αP to the signer; the signer then computes $e(S_{ID}, \alpha P)$. If the equation $e(S_{ID}, \alpha P) = e(H_1(ID, rP), P_{pub})^{\alpha}$ holds, i.e., ID corresponds to rP

and $r'P$, the arbiter deduces that PKG is dishonest because the master-key s is only known to PKG.

We have the following security results for our scheme. The proof can be seen in the full version of this paper.

Theorem 1. The proposed scheme is blind.

Theorem 2. The proposed scheme is secure against adaptively chosen message and ID attack in the random oracle, assuming the hardness of CDHP.

3 ID-Based Proxy Blind Signature without Trusted PKG

Our ID-based proxy blind signature scheme without trusted PKG is based on Cha-Cheon's ID-based signature scheme [3], Boneh et al.'s pairing-based short signature scheme [2] and Zhang-Kim's ID-based blind signature scheme [6].

Define four cryptographic hash functions $H_1 : \{0,1\}^* \times G_1 \to G_1$, $H_2 : \{0,1\}^* \times G_1 \to Z_q$, $H_3 : \{0,1\}^* \to G_1$ and $H_4 : \{0,1\}^* \times G_1 \to Z_q$.

Setup: PKG chooses $s \in_R Z_q^*$ and sets $P_{pub} = sP$. PKG publishes system parameters $params := \{G_1, G_2, e, q, P, P_{pub}, H_1, H_2, H_3, H_4\}$ and keeps s as the master-key.

Extract: Let Alice be the original signer with ID_A and Bob be the proxy signer with ID_B. Alice chooses $r_A \in_R Z_q^*$ as her partial private key and sends $r_A P$ to PKG. PKG computes $Q_A = H_1(ID_A, r_A P)$ and $S_A = sQ_A$. Bob chooses $r_B \in_R Z_q^*$ as his partial private key and sends $r_B P$ to PKG. PKG computes $Q_B = H_1(ID_B, r_B P)$ and $S_B = sQ_B$. Alice's private key is (S_A, r_A) and Bob's private key is (S_B, r_B).

Sign: Alice makes a message m_ω, selects $v \in_R Z_q$, and computes $U_\omega = vQ_A$, $h_\omega = H_2(m_\omega, U_\omega)$, $V_\omega = (v + h_\omega)S_A$ and $T_\omega = r_A H_3(m_\omega)$.

The signature on m_ω is the warrant $W = (U_\omega, V_\omega, T_\omega, m_\omega, r_A P)$.

Verify: Verifier computes $h_\omega = H_2(m_\omega, U_\omega)$ and checks if

$$e(V_\omega, P) = e(U_\omega + h_\omega Q_A, P_{pub}),$$
$$e(T_\omega, P) = e(H_3(m_\omega), r_A P).$$

If it is right, he accepts it, and rejects it otherwise.

Proxy-Designate: In order to designate Bob as a proxy signer, Alice sends Bob an appropriate warrant W. Bob verifies this signature and if it is valid, he compute the proxy secret key $S_P = h_\omega S_B + V_\omega$ and $r_P = T_\omega + r_B H_3(m_\omega)$.

Proxy Blind Sign: Suppose that m is the message to be signed.

- Bob chooses $t \in_R Z_q^*$ and $t' \in_R Z_q^*$, computes $U = t(h_\omega(Q_A + Q_B) + U_\omega)$ and $X = t' H_3(m_\omega)$, and sends U, X to the user as a commitment.

Bob also sends $r_A P, r_B P, m_\omega, U_\omega, h_\omega$ to the user.

- (**Blinding**) The user chooses $\alpha, \beta \in_R Z_q^*$ as blinding factors, computes $R = \alpha U + \alpha\beta(h_\omega(Q_A + Q_B) + U_\omega)$ and $h = \alpha^{-1}H_4(m, R) + \beta$, sends h to the signer.

The user chooses $\xi, \eta \in_R Z_q^*$ as blinding factors, computes $C = \xi X + \xi\eta H_3(m_\omega)$ and $k = \xi^{-1}H_4(m, C) + \eta$, sends k to the signer.

- (**Signing**) Bob sends back V and Y, where $V = (t + h)S_P$ and $Y = (t' + k)r_P$.

- (**Unblinding**) The user computes $S = \alpha V$ and $D = \xi Y$.

Then $(m_\omega, U_\omega, R, S, C, D, ID_B, r_A P, r_B P)$ is the proxy blind signature of m.

Proxy Blind Verify: Verifier computes $Q_A = H_1(ID_A, r_A P)$, $Q_B = H_1(ID_B, r_B P)$, $h_\omega = H_2(m_\omega, U_\omega)$, and accept the signature if and only if

$$e(S, P) = e(R + H_4(m, R)(h_\omega(Q_A + Q_B) + U_\omega), P_{pub}),$$

$$e(D, P) = e(C + H_4(m, C)H_3(m_\omega), r_A P + r_B P).$$

As in Section 2, if PKG impersonates an honest signer to sign a document, the signer can prove with the secret information "r" that the PKG is dishonest.

We have the following security result for this scheme. The proof can be seen in the full version of this paper.

Theorem 3. The proposed scheme is secure against adaptively chosen message and ID attack in the random oracle, assuming the hardness of CDHP.

4 Conclusions

In this paper, we proposed an ID-based blind signature and proxy blind signature without trusted PKG. Our schemes can avoid the inherent key escrow problem.

References

[1] Boldyreva, A.: Efficient threshold signature, multisignature and blind signature schemes based on the Gap-Diffie-Hellman-group signature scheme. In: Desmedt, Y.G. (ed.) PKC 2003. LNCS, vol. 2567, pp. 31–46. Springer, Heidelberg (2002)

[2] Boneh, D., Lynn, B., Shacham, H.: Short Signatures from the Weil Pairing. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 514–532. Springer, Heidelberg (2001)

[3] Cha, J., Cheon, J.: An Identity-Based Signature from Gap Diffie-Hellman Groups. In: Desmedt, Y.G. (ed.) PKC 2003. LNCS, vol. 2567, pp. 18–30. Springer, Heidelberg (2002)

[4] Chen, X., Zhang, F., Kim, K.: A New ID-Based Group Signature Scheme from Bilinear Pairings, Cryptology ePrint Archive, Report 2003/116, http://eprint.iacr.org/2003/116/

[5] Shamir, A.: ID-Based Cryptosystems and Signature Schemes. In: Blakely, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985)

[6] Zhang, F., Kim, K.: Efficient ID-based blind signature and proxy signature from bilinear pairings. In: Safavi-Naini, R., Seberry, J. (eds.) ACISP 2003. LNCS, vol. 2727, pp. 312–323. Springer, Heidelberg (2003)

The Combination of CMS with PMC for Improving Robustness of Speech Recognition Systems

Hadi Veisi and Hossein Sameti

Department of Computer Engineering, Sharif University of Technology, Tehran, Iran
veisi@ce.sharif.edu, sameti@sharif.edu

Abstract. This paper addresses the robustness problem of automatic speech recognition systems for real applications in presence of noise. PMCC algorithm is proposed for combining PMC technique with CMS method. The proposed algorithm utilizes the CMS normalization ability in PMC method to takes the advantages of these methods to compensate the effect of both additive and convolutional noises. Also, we have investigated VTLN for speaker normalization and MLLR and MAP for speaker and acoustic adaptation. Different combinations of these methods are used to achieve robustness and making the system usable in real applications. Our evaluations are done on 4 different real noisy tasks on Nevisa recognition system. Experimental results show the effectiveness of proposed method and significant improvement in system performance.

Keywords: Speech Recognition, Robustness, Parallel Model Combination (PMC), Cepstral Mean Subtraction (CMS).

1 Introduction

Now is the time of moving speech recognition systems from laboratories to real-world applications. The most challenging problem in this migration is the effect of new environment in the performance degradation of these systems. The problem of performance reduction is due to the mismatch between train and test acoustic conditions. There are several approaches to overcome the problem. One main category of robustness methods such as Cepstral Mean Subtraction (CMS) [1] and Vocal Tract Length Normalization (VTLN) [1], [3], [4] tries to extract inherent robust features that we have employed them in this paper. Model compensation methods are another category that attempt to modify pattern-matching stage in such a way that decreases the mismatch between models and new environment. Parallel Model Combination (PMC) [1], [2] is one of these techniques that estimates the noisy models. In this paper a novel method is presented to use PMC in systems with CMS normalization ability. CMS is a powerful method for removing the effect of convolutional noises and it is widely used in operational speech recognition systems. On the other hand, PMC is often used for compensating the additive noise effects. The combination of PMC with CMS enables speech recognition systems to utilize the benefits of both methods and can overcome both additive and convolutional noise. The hindering problem in combining these two was that PMC algorithm required invertible processes in the front-end of the system and the CMS normalization is not an invertible process. The

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 825–829, 2008.

method presented here overcomes this problem. Maximum Likelihood Linear Regression (MLLR) [6], [7] and Maximum A Posteriori (MAP) [7] are popular model adaptation methods that are evaluated and combined with proposed method in this paper.

The proposed algorithm to integrate PMC with CMS is presented in Section 2. Section 3 describes the framework of experiments and shows results of our examinations and Finally, Section 4 gives a brief summary and conclusion.

2 The Combination of PMC with CMS

Parallel Model Combination (PMC) [2] attempts to estimate the model parameters of the noisy speech signal from the parameters of clean model and noise model. In this paper we handle additive noise compensation using this method. The basic assumption behind PMC is the independence of speech and noise signals and their linear combination in spectral domain. The acoustic models are mixtures of Gaussian distributions in MFCC or so-called cepstral domain in most modern speech recognizers which can be characterized by a set of means and covariance matrices $\{\mu^{cep}, \Sigma^{cep}\}$. By having models of both clean speech and environmental noise, the estimation of the noisy model parameters is done by combining noise and clean parameters in linear spectral domain as in following steps:

1. Mapping the clean and noise parameters from cepstral to linear spectral domain.
2. Combine the clean and noise parameters linearly to estimating noisy parameters.
3. Mapping back the noisy parameters from linear spectral to cepstral domain.

Cepstral Mean Normalization (CMN), on the other hand, is a well-known and semi-standard normalization method to cure channel distortion effect. It is also known as CMS since the normalization is actually done by subtracting the time average of coefficients from them. CMS is very effective for compensation of convolutive noises [1]. Also, researches have been mainly done on using PMC for additive noise compensation. PMC and CMS are both effective and applicable separately on practical speech recognition systems but using both of the methods in a system is not straight forward. PMC assumes that all blocks in signal parameterization with MFCC are invertible to map parameters form cepstral domain into linear domain and vice versa but the CMS normalization is not an invertible process.

Here, an algorithm is proposed to solve this problem and enable us to utilize the benefits of both PMC and CMS methods. The first step for this is to make CMS invertible. To do this, we use an approximation of CMS and a fixed normalization vector for CMS normalization instead of the computing a separate normalization vector for each sentence. This fixed normalization vector, $\overline{O}^{cep}$, which is the average of cepstral vectors of the whole training data is subtracted form each feature vector. Using this assumption, the algorithm of using the PMC with CMS (PMCC) is presented below.

1. CMS Inversion- add $\overline{O}^{cep}$ to mean vectors of clean models, $\mu^{cep} = \mu^{cep-cms} + \overline{O}^{cep}$.
2. DCT Inversion- This is done for clean and noise parameters and $\overline{O}^{cep}$.
3. Log Inversion- This is done for clean and noise parameters and $\overline{O}^{\log}$.

4. The combination of:
   4.1. Clean $\{\mu^{lin}, \Sigma^{lin}\}$ and noise $\{\tilde{\mu}^{lin}, \tilde{\Sigma}^{lin}\}$ parameters in linear domain.
   4.2. The average vector $\overline{O}^{lin}$ with the mean of noise $\tilde{\mu}^{lin}$, $\hat{\overline{O}}^{lin} = g.\overline{O}^{lin} + \tilde{\mu}^{lin}$.
5. Return the estimated parameters $\hat{\overline{O}}^{lin}$ and $\{\hat{\mu}^{lin}, \hat{\Sigma}^{lin}\}$ back to the cepstral domain.
6. Perform CMS on the estimated noisy mean vectors, $\hat{\mu}^{cep-cms} = \hat{\mu}^{cep} + \hat{\overline{O}}^{cep}$.

In this algorithm, only mean parameters are estimated because the CMS normalization dose not affects the covariance matrix. Also, it is assumed that the noise is modeled with one Gaussian in one state in cepstral domain and $\overline{O}^{cep}$ is extracted in cepstral domain. If the environment noise is modeled by more than one Gaussian, then the average of noise should be computed in another way i.e. like $\overline{O}^{lin}$ and combined in step 4.2 with $\overline{O}^{lin}$. Another way to estimate the noisy mean vector, $\hat{\overline{O}}^{cep}$, is to extract the average of all frames of the noisy test set. If so, the estimated noisy values $\hat{\overline{O}}^{cep}$, can be ignored.

3 Experiments

In evaluation of the robustness methods and proposed PMCC algorithm, Nevisa Persian recognition system [1], [7] is used. Nevisa is a HMM-based recognition system with MFCC front-end. In these experiments each model contains 6 states and each state has 16 Gaussian mixtures with diagonal covariance matrixes. In pre-processing of the base-line system 12 cepstral coefficients with first and second derivations are used as features for each frame. The frames are extracted from 20ms signals with 12ms overlap. In the baseline system, 30 Persian phonemes are modeled using FarsDat database [1] and 1.1K lexicon with bi-gram language model is used. To evaluate the performance of the robustness methods, 4 real noisy data sets [1], [7] are used. Each of these sets is uttered by 5 males and 2 females. Also each set includes two subsets identified as adaptation subset (contains 175 sentences) and test subset (consists of 140 sentences) [7]. Each task demonstrates a new environment which differs from training environment. Tasks A and B are recorded in office environment with condenser and dynamic microphones and their average SNR are 12dB and 30dB, respectively. Both tasks C and D are recorded with condenser microphone in office environment and in presence of exhibition and car noises and the corresponding SNRs of these tasks are 9dB and 7dB, respectively. In these experiments, word error rates (WER) is used as evaluation criteria. The WER of system is 18.3% on clean test set but for the noisy tasks the WER increases dramatically.

To model the noise of each task, silence segments of the test sentences in each task are used. The noise models are single Gaussian and single state. We have used both Log-Normal (LN) and Log-Add (LA) [2] approximations for PMC adaptation. Table 1 shows the results of baseline system (i.e. recognition using clean models) and robustness methods on 4 noisy tasks. As the results show, performance of the system is extremely deficient in presence of noise. The improvements resulted using PMC approximations are completely noticeable on all tasks and particularly on task C and D. This is because the Log-Add and Log-Normal model the additive noise naturally and

compensate the effect of this type of noise. In task A and B there is very little background noise and the main cause of degradation of the performance are other mismatch sources such as speaker and microphone variations. Comparing the results of LN and LA methods, shows that the effect of covariance matrix (here the variance) adaptation in PMC method is not significant. The forth row in table 1 shows the results of proposed PMCC (use Log-Add here) for mean adaptation in this evaluation. This method provides apparent improvement in the performance on all tasks. The relative improvements on tasks A and B are more than two other tasks. This is because of the effectiveness of CMS normalization on tasks A and B.

Table 1. WER (%) obtained by robustness methods, PMCC and combination of them on 4 noisy tasks

Method	***Task A***	***Task B***	***Task C***	***Task D***
No Robustness	74.04	75.32	121.65	107.8
LN	55.88	62.28	83.7	64.96
LA	55.06	62.63	85.1	66.24
PMCC	*47.73*	*49.13*	*82.24*	*60.93*
VTLN	67.77	67.39	118.28	104.85
VTLN+MLLR	30.37	32.87	82.52	60.07
LA+MLLR	35.72	45.45	86.87	72.57
LA+MAP	36.32	41.91	64.84	57.36
LA+VTLN+MLLR	***28.03***	***31.32***	73.64	64.63
LA+VTLN+MAP	38.09	46.80	70.11	55.36
PMCC+MLLR	*31.80*	*35.77*	***61.50***	***51.68***

MLLR and MAP are two effective methods which have been employed in most practical systems and they are used along with VTLN speaker normalization in this research. As the results show, VTLN is effective only in clean and less-noise-dominant conditions, i.e. tasks A and B. This is possibly because of wrong estimation of signal's spectrum and wrong frame-state alignment in noisy conditions. Using both VTLN and MLLR adaptation methods results in better performance due to the additive effect of these two methods [6]. Taking the advantages of PMC (Log-Add method here) noise compensation with adaptation ability of MLLR/MAP, results more WER improvement. Combining the normalization, adaptation and noise compensation abilities is realized by using VTLN, MLLR/MAP and PMC methods together. In these experiments, the VTLN normalized models are noise compensated with LA and then adapted with MLLR/MAP. The results of these experiments are considerably successful particularly on the first two tasks. As mentioned, VTLN and the combination of it with other methods on noise dominant tasks do not improve the performance. On the other hand, using LA, VTLN and MLLR methods provide the best results on task A and B which is due to additive effect of these methods in less-noisy environments. Also, as the results show, adaptation using MAP achieves better results on noise-dominant tasks in comparison with MLLR. This may be because of the linearity coverage limitation of MLLR. Finally, using the proposed PMCC method with the MLLR adaptation ability yields remarkable improvements in comparison with LA+MLLR results and provides the best results on more noisy tasks, C and D.

The results of using PMCC with other methods also demonstrate the compatibility of it with these robustness methods.

4 Summary and Conclusion

In his paper, we presented PMCC algorithm for improving the noise compensation capability of PMC in real applications. PMCC integrates CMS with PMC which enables us to take the channel compensation advantages of CMS beside additive noise adaptation ability of PMC. Unlike the most already reported works on PMC-based robustness, we focused on the robustness of Persian continuous speech recognition system for real applications and did our evaluations on 4 different real noisy conditions. PMCC algorithm results evident improvement in comparison to standard PMC in all noisy tasks and especially on tasks with more channel distortion effect. Also VTLN speaker normalization and MLLR and MAP adaptation methods were examined. The robustness abilities of these methods in combination with each other and in combination with PMCC were experienced. The combination of these robustness methods with our proposed method results in more robustness too.

References

1. Veisi, H.: Model-based methods for noise robust speech recognition systems., M.S thesis, Computer department, Sharif University of Technology (November 2005)
2. Gales, M.J.F.: Model-Based Techniques for Noise Robust Speech Recognition., Ph.D. thesis, University of Cambridge (September 1995)
3. Veisi, H., et al.: Improving the Robustness of Persian Large Vocabulary Continuous Speech Recognition System for Real Applications. In: IEEE International Conference on Information & Communication Technologies, from Theory to Applications (ICTTA), Syria (April 2006)
4. Zhan, P., Westphal, M., Finke, M., Waibel, A.: Speaker Normalization and Speaker Adaptation – a Combination for Conversational Speech Recognition. In: Eurospeech Conference, Greece (1997)
5. Yamamoto, H., et al.: Fast Speech Recognition Algorithm under Noisy Environment Using Modified CMS-PMC and Improved IDMM+SQ. In: ICASSP 1997 (1997)
6. Leggetter, C.J., Woodland, P.C.: Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density HMMs. Computer Speech and Language 9, 171–186 (1995)
7. Hosseinzadeh, K.: Improving the Accuracy of Continuous Speech Recognition in Noisy Environments M.S thesis, Computer dep., Sharif University of Technology (November 2004)

A Dynamic Multi Agent-Based Approach to Parallelizing Genetic Algorithm

Mohsen Momeni and Kamran Zamanifar

Department of Computer Engineering, University of Isfahan, Isfahan, Iran
{mmomeni,zamanifar}@eng.ui.ac.ir

Abstract. Multi agent systems have been widely used to parallelizing processes in network-based computation environments. Parallelization of computational load could improve the performance of genetic algorithms and decrease their computational time. In this paper, a new distributed multi-agent system has been proposed for parallelizing genetic algorithm in usual networks. This dynamic approach could improve the performance of parallel genetic algorithm in a network-based context. We also provide a new migration policy for proposed architecture which leads us to attain a better performance for Parallel Genetic Algorithm. Experimental results show the efficiency of proposed method comparing to previous studies.

Keywords: Distributed Multi Agent System, Parallel Genetic Algorithm, Mobility, Evolutionary Computing, Distributed System.

1 Introduction

Parallelizing genetic algorithms usually considered as one of the best approaches that could decrease computational time in these algorithms and performing them in a reasonable time. Distribution of the population also could affect on convergence and improves the fitness of solutions better than the simple global (using one population) genetic algorithm. Running an Island Parallel Model of genetic algorithm, even on a single processor (without the parallelism) is often more effective than running a single population evolutionary algorithm with the same cumulative population size [9]. But we need a migration policy to transfer some individuals from one island to another to keep their diversity. Diversity in the population prevents it from standing near local optima and early convergence. Dynamic changes in computational network must be considered in parallelizing genetic algorithm on a network-based architecture. The multi-agent system seems to be an attractive way to overcome transparency and scalability problems in distributed computing systems [7].

We propose a new agent-based approach that uses an investigated migration policy which improves parallel genetic algorithm performance comparing to previous agent-based approaches. This paper is organized as follows: In section 2 we review the background of this research. The architecture has been described in section 3. Implementation and experimental results have been represented in section 4 and finally we conclude our paper in section 5.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 830–833, 2008.

2 Related Works

We can recognize four major types of parallel genetic algorithms [2]: *Master-slave, Coarse-grain, Fine-grain, and hierarchical* parallel genetic algorithms. High communication rate in fine grain model and central architecture and synchronization in master-slave model, which are the fundamental characteristics of these approaches, prevent these models to attain good performance in a network-based parallelization environment. Network architecture has very similarities with the MIMD architecture, and it seems that coarse grain model is the best profit one to be used in a network-based parallelization of genetic algorithms. There is some agent-based parallelization of evolutionary algorithm in the literature which proposed a notable approaches in this kind of parallelization [4], [5], [6], [7], and [8]. Common proposed models use a coarse-grain based model for development agent-based parallel evolutionary algorithms. Among all these models and platforms referenced above, only two models use the mobility of agent to improve performance in genetic algorithms. Momot proposed a multi-agent system in which when utilization of a host by other applications become high, the population of agent divides in to two populations and one of them leaves the host [7]. The other system that uses mobility property of agents in his model is reported in Lee Approach [6]. In this model, which has some similarities with our proposed model, when an agent find out that the computation of evolutionary approach isn't possible in the current environment, it leaves it and goes to a new environment. But Lee's method uses general synchronization and uses a central manager agent. This central agent could become a bottleneck for the system and decrease the scalability of the model. In our proposed approach we try to attain a full-distributed peer-to-peer system that doesn't have these mentioned problems.

3 Proposed Multi-agent Architecture

In our approach there are two kinds of agents: Static agents located in each host and don't move over the process of evolution. These agents task is to recognize whether a context is proper to do evolutionary operations and evolution process or not. We called them status agents (SA) in this approach. Other agents that we called them computation agents could be moved over the process of evolution respect to their environment conditions. Fig. 1 represents an overall schematic of this framework.

The process of evolution in each agent has organized in some two-phase-based generations that we called them *epoch*. Each epoch has two phases: Computation phase and Migration phase. In computation phase each agent works separately from other agents and apply genetic operations on its population for a predefined number of generations. After this phase it exchanges its migrants with its neighbors. The process begins with sending messages from *initiator agent* to the status agents. Then initiator agent defines a primitive topology for computation agents according to average load of hosts, produces them and organizes a virtual cube between these agents. Then it sends them to their predefined context and informs them that where their neighbors are. These computation agents find their seats and communicate their neighbors with their messages. While an agent works on a host, it is probable that other applications starts on machine and their computational load prevents the agent evolution process or increases its computational time in a way that it cannot coordinate its evolution process with the others. Status agent is responsible to send the current status

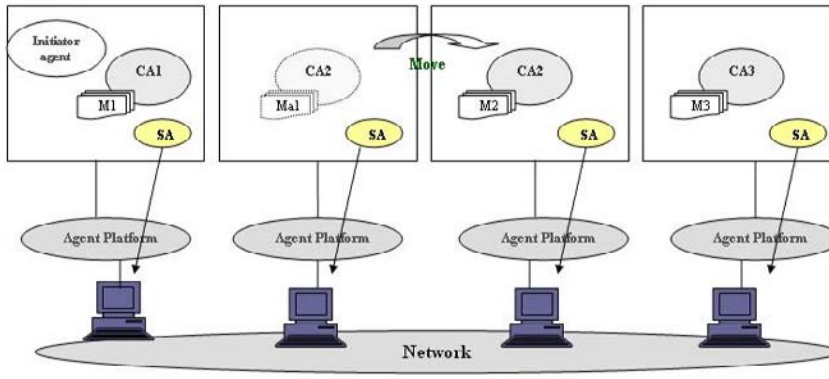


Fig. 1. Proposed multi-agent architecture

properties to computation agent in each predefined period of time. If the average utilization of the host with other applications is higher than a threshold in a period of time, computation agent decides to move from this host to another idle host.

In semi-asynchronous migration, our proposed approach for migration, we try to prevent our populations from pitfalls that could happen to asynchronous types, but without high synchronization that blocks all agents until termination of the slowest agent computations. In our approach, computation agents wait rarely and there is not a long blockage time after each epoch. Also it prevents agents from maverick evolution and incompatible migrations. When evolutionary phase of an epoch terminates in one computation agent, it sends a request to receive next epoch migrants to its migration agent. If one or more of its neighbor agents terminates their computational jobs in this epoch and sends their migrants in advance, it uses a selected collection of its neighbor migrants as this epoch migrants. But it doesn't use next epoch migrants. These migrants have been saved in migration agent, until it uses them. Only when none of the agent's neighbors has not been terminated their computational step faster than the agent, will the agent wait until the arrival of the first deme of the migrants of this epoch from its fastest neighbor and use its migrants lonely.

4 Implementation and Experimental Results

We use the Aglet framework [10] to develop our agents. The experiments were conducted within a network of 13 PC machines connected by a TCP/IP protocol based network. Machines ranged from PIV 1700GHz, 256Mb RAM up to DuoCore 3000GHz, 1Gb RAM machines. We define the number of computation agents to 8 in this implementation and these agents form a virtual cube among them, and each of them has 3 neighbors. We have used DeJong's test suite functions, QAP Problem, and SC problem that were mentioned in [9], and a time consuming problem, Air Quality Optimization which have been described in [1] to test our approach and compared its results with previous methods. We perform computations for four approaches: Lee-Method [6], Synchronous distributed, asynchronous distributed and semi-asynchronous distributed models. Each column in table 1 contains values gained from

each tested method in finding predefined quality solution of this problem. As it can be seen, semi-asynchronous approach has shown better results in time consuming "Air Quality Optimization" test problem.

Table 1. Results of proposed approach comparing with Lee, Sync and Async approaches

Test Func.	Lee Method	Synchron Dist.	Async Dist.	Semi-Async Dist.
DeJongTests	00.00.0.96	00.00.00.99	00.00.00.84	00.00.00.97
QAP	00.12.20.83	00.13.44.63	00.10.33.44	00.13.00.05
SCP	00.01.28.53	00.01.29.34	00.01.23.86	00.01.21.35
AQO	01.43.67.78	01.27.34.09	01.29.34.20	01.21.40.38

5 Conclusions and Future Works

In this paper we proposed a new dynamic multi agent-based parallel genetic algorithm that could improve the performance of genetic algorithm in heterogeneous and ever-changing networks. We have proposed, semi-asynchronous approach, a new migration policy that increases the performance of the system, comparing with synchronous and asynchronous migration policies. We are going to extend our approach to provide a high performance internet-based distributed genetic algorithm and study its performance on the internet environment.

References

1. Baugh, W., Kumar, S.V.: Asynchronous Genetic Algorithm for Heterogeneous Network Using Coarse-Grained Dataflow. In: GECCO 2003, pp. 730–741 (2003)
2. Cantu-Paz, E.: Efficient and accurate parallel genetic algorithms. Kluwer Academic Publishers, Dordrecht (2000)
3. James, T., Barkhi, R., Johnson, J.: Platform impact on performance of parallel genetic algorithms: Design and implementation considerations. Engineering Applications of Artificial Intelligence 19, 843–856 (2006)
4. Kim, J.: Distributed Genetic Algorithm with Multiple Populations Using Multi-Agent. In: Zima, H.P., Joe, K., Sato, M., Seo, Y., Shimasaki, M. (eds.) ISHPC 2002. LNCS, vol. 2327, pp. 329–334. Springer, Heidelberg (2002)
5. Kisiel-Dorohinicki, M.: Agent-based Models and Platforms for Parallel Evolutionary Algorithms. In: Bubak, M., van Albada, G.D., Sloot, P.M.A., Dongarra, J. (eds.) ICCS 2004. LNCS, vol. 3038, pp. 646–653. Springer, Heidelberg (2004)
6. Lee, W.P.: Parallelizing evolutionary computation: A mobile agent-based approach. Expert systems with Applications 32, 318–328 (2007)
7. Momot, J., Kosacki, K., Grochowski, M., Uhruski, P., Schaefer, R.: Multi-Agent System for Irregular Parallel Genetic Computations. In: International Conference on Computational Science, pp. 623–630 (2004)
8. Murata, Y., Shibata, K., Yasumoto, K., Ito, M.: Agent Oriented Self Adaptive Genetic Algorithm. In: CCN, Cambridge, USA, November 4-6, pp. 348–353 (2002)
9. Whitley, D.: An Overview of Evolutionary Algorithms: Practical Issues and Common Pitfalls. Information and Software Technology 43, 817–831 (2001)
10. Aglets (IBM Agent Development Kit), `http://www.trl.ibm.com/aglets/`

Fuzzy Neighborhood Allocation (FNA): A Fuzzy Approach to Improve Near Neighborhood Allocation in DDB

Reza Basseda[1], Maseud Rahgozar[2], and Caro Lucas[2]

[1] Database Research Group, School of Electrical and Computer Engineering, Faculty of Engineering, University of Tehran, North Karegar Ave., Tehran, Iran
R.Basseda@ece.ut.ac.ir
[2] Database Research Group, Control and Intelligent Processing Center of Excellence, School of Electrical and Computer Engineering, Faculty of Engineering, University of Tehran, North Karegar Ave., Tehran, Iran
Rahgozar@ut.ac.ir, Lucas@ipm.ir

Abstract. Allocating data fragments in distributed database systems is an important issue in distributed database (DDB) systems. In this paper, we are going to improve the effectiveness of current NNA algorithm using a Fuzzy inference engine. Results indicate that, our fuzzy based NNA algorithm leads 5% gain in some of systems performance metrics. This algorithm, providing a data clustering mechanism, which is very suitable for DDBS in the networks, with heavy traffic loads, and frequent data access requests.

1 Introduction

Developments using databases and networking technologies in the past two decades have led to important advances in distributed database systems (DDB). The objective of a data allocation algorithm is to determine the assignment of fragments at different sites so as to minimize the total data transfer cost incurred in executing a set of queries. This is equivalent to minimizing the average query execution time, which is of primary importance in a wide class of distributed conventional or multimedia database systems. Initial studies on dynamic data allocation give a framework for data redistribution and demonstrate how to perform the redistribution process in a minimum possible time. In [3] a dynamic data allocation algorithm for non-replicated database systems named optimal algorithm is proposed without any modeling to analyze the algorithm. In [4] a threshold algorithm is proposed for dynamic data allocation which dynamically reallocates data with respect to data access patterns. It focused on load balancing issues. Near Neighborhood Allocation (NNA) is a variation of the optimal algorithm. It is different with optimal algorithm in choosing destination of migrating fragment. In NNA, destination of moved fragment is the neighbor of the source site, which is in the path from the source to the site with highest access pattern. This algorithm avoids moving data fragment too frequently [2].

All of above algorithms using crisp method to move data along network paths. Estimating time and place (destination) of a data fragment depends on various parameters

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 834–837, 2008.

such as access pattern, bandwidth of network links and etc. They do not prevent an oscillation of data in case of frequent changes in data access pattern.

In this paper we are going to present and analyze a new approach for dynamic data allocation algorithm named Fuzzy Neighborhood Allocation. This algorithm is based on NNA [2], but with different strategy for selecting nodes for data movements and different strategy in migrating fragments. Proposed algorithm uses fuzzy method to avoid oscillation condition and prevent redundant data fragment migration.

The rest of this paper is organized as following: In next section we will describe our method, third section presents our simulation environment and its results and last section gives our conclusion.

2 Methodology

As mentioned above, detecting oscillation is important in DDB. Rapid changing of access pattern for a single fragment may cause problems in DDB system. Fragment migration between two sites leads high delay in accessing fragment. Migrating fragment is inaccessible during migration because fragment is locked when it moves from one site to another. We are going to solve this problem by avoiding oscillations. We consider access pattern of a site and recognize oscillation condition through differentiation of access pattern. After degrading of access pattern with using mean factor, we use a fuzzy *AND* operator between smoothed access pattern and fuzzy compliment of differentiation of access pattern. The result show revised access pattern, which can be used in deciding of fragment migration. We trace access counts in 20 time sluts. Each time slut comprises of 50-clock cycle. Access pattern of site j for fragment i during the time slut is calculated as below:

$$P_{ijt} = \frac{S_{ijt}}{\sum_{K \in Sites} S_{ikt}} \tag{1}$$

Smoothed value of this pattern is calculated as below:

$$Smoothed(P_{ijt}) = \frac{P_{ijt} + P_{ij(t+1)}}{2} \tag{2}$$

Comparison of these values for a sample data is showed in figure1. Differentiation of accessed patterns are calculated as:

$$\frac{dP_{ijt}}{dt} \approx D_{ijt} = \frac{P_{ij(t+1)} - P_{ijt}}{\Delta t} \tag{3}$$

Absolute value of differentiation fuzzified.

Finally, revised accessed pattern is calculated through fuzzy and operation as bellow:

$$R_{ijt} = Smoothed(P_{ijt}) \wedge \sim D_{ijt} \tag{4}$$

Revised access pattern is defuzzified and used to compare with revised access pattern of owner. Decision for fragment migration is made according to these values.

Now, we use the link state algorithm. We are going to choose a proper link to move data fragment. We must compare links of owner. Each link is evaluated by the sites, which the link is in their access using link state algorithm. We evaluated the sites by compliment of fuzzified distance vector multiplied by fuzzified access request vector. Using a fuzzy *OR* operation, we conclude these parameters for a single link. Suppose N is the set of nodes which are connected to the owner via l in Dijkstra SPF algorithm. Then we will have:

$$W_l = \bigcup_{j \in N} V_j \tag{5}$$

Then, after defuzzification phase, we can compare links. Figure 1 shows decision-making system schema.

3 Experimental Evaluation and Simulation Results

We developed software to simulate FNA algorithm and compare it with NNA. This simulator is configurable for testing different network topologies and different data requests and/or allocation conditions. In our simulator, we mark each packet's send and receive time. Using timestamps, we could compare algorithms for different factors. Detailed information regarding the implementation of this software is available in [5] and [6].

So, our system has 9 nodes as shown in [6]. We used IKCo guarantee data as our benchmark. In this section we considered each dealer as an independent site and each car referring as a requesting transaction.

In our experiments, we consider average delay for receiving the response (response time) for a fragment request. We will investigate the effect of different fragment size on this factor. According to Figure 1, for small fragments the average time spent for moving data in FNA algorithm is larger than NNA and for larger fragments this is reversed. The reason is that for small fragments the cost of moving data to destination node is low and so, the movement cost does not exceed the access cost. In the case of large fragments, the movement of fragments takes more time and also increases the network traffic.

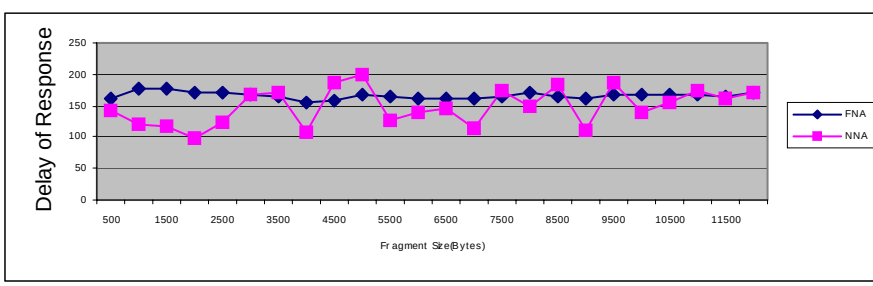


Fig. 1. Effect of fragment size on the delay of responding to a fragment request in optimal approach and in NNA

4 Conclusion

In this study we represented a new dynamic data allocation algorithm in DDB environment named Fuzzy Neighborhood Allocation (FNA) algorithm. Findings of our experiments indicated that, the FNA algorithm performs better for larger fragment size, but for small fragment size, the NNA and optimal algorithm performs better.

Here, we just studied these algorithms on non-replicated distributed database systems. Further studies are needed to test FNA, NNA and optimal algorithms in replicated distributed database systems.

References

[1] Berkan, R.C., Trubatch, S.L.: Fuzzy Systems Design Principles. IEEE Press, New York (1997)

[2] Basseda, R., Tasharofi, S., Rahgozar, M.: Near Neighborhood Allocation (NNA): A Novel Dynamic Data Allocation Algorithm in DDB. In: Proceedings of 11th Computer Society of Iran Computer Conference (CSICC 2006), Tehran (2006)

[3] Brunstroml, A., Leutenegger, S.T., Simhal, R.: Experimental Evaluation of Dynamic Data Allocation Strategies in a Distributed Database with changing Workloads. ACM Transactions on Database Systems (1995)

[4] Ulus, T., Uysal, M.: Heuristic Approach to Dynamic Data Allocation in Distributed Database Systems. Pakistan Journal of Information and Technology 2(3), 231–239 (2003)

[5] Basseda, R., Tasharofi, S.: Design and Implementation of an Environment for Simulation and Evaluation of Data Allocation Models in Distributed Database Systems, Technical Report No. DBRG.RB-ST.A50701 (2005)

[6] Basseda, R., Tasharofi, S.: Data Allocation in Distributed Database Systems, Technical Report No. DBRG.RB-ST.A50715 (2005)

OPWUMP: An Architecture for Online Predicting in WUM-Based Personalization System

Mehrdad Jalali[1], Norwati Mustapha[2], Md Nasir B. Sulaiman[2], and Ali Mamat[2]

[1] Software Engineering Department, Islamic Azad University of Mashhad, Iran
mehrjalali@gmail.com
[2] Department of Computer Science, Faculty of Computer Science and IT
Putra University of Malaysia Selangor, Malaysia
{norwati,nasir,ali}@fsktm.upm.edu.my

Abstract. The Internet is one of the fastest growing areas of intelligence gathering. During their navigation web users leave many records of their activity. This huge amount of data can be a useful source of knowledge. Sophisticated mining processes are needed for this knowledge to be extracted, understood and used. Web Usage Mining (WUM) systems are specifically designed to carry out this task by analyzing the data representing usage data about a particular Web Site. WUM can model user behavior and, therefore, to forecast their future movements. Online prediction is one web usage mining application. However, the accuracy of the prediction and classification in the current architecture of predicting users' future requests systems can not still satisfy users especially in Huge Web sites. To provide online prediction efficiently, we develop an architecture for online predicting in WUM-based personalization system (OPWUMP).This article advances an architecture of Web usage mining for enhancing accuracy of classification by interaction between classification, evaluation, current user activates and user profile in online phase of this architecture.

Keywords: Web Usage Mining, classification, online prediction.

1 Introduction

With the explosive growth of knowledge available on the World Wide Web, which lacks an integrated structure or schema, it becomes much more difficult for users to access relevant information efficiently. Meanwhile, the substantial increase in the number of websites presents a challenging task for webmasters to organize the contents of the websites to cater to the needs of users. Modeling and analyzing web navigation behavior is helpful for understand what information online users demand. Following that, the analyzed results can be seen as knowledge to be used in intelligent online applications, refining web site maps, and improving searching accuracy when seeking information. Nevertheless, an online navigation behavior grows each passing day, and thus extracting intelligently from it is a difficult issue. Web Usage Mining (WUM) is the process of extracting knowledge from Web user's access data by exploiting Data Mining technologies [4].It can be used for different purposes such as personalization, system improvement and site modification. Typically, the WUM

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 838–841, 2008.

prediction process is structured according to two components performed online and off-line with respect to the Web server activity [1], [2] and [5]. The off-line component is aimed at building the knowledge base by analyzing historical data, such as server access log files, that is then used in the online component.

In this paper we will focus on the architectural issues of WUM systems, by proposing an online prediction system. In this system we have propose an architecture for improving accuracy of prediction and classification. The rest of this paper is organized as follows: In section 2, we review recent research advances in web usage mining. Section 3 describes the architecture of proposed system for online predicting. Finally, section 4 summarizes the paper and introduces future work.

2 Background and Related Works

Recently, several WUM systems have been proposed to predicting user's preference and their navigation behavior [3] and [1]. One of the latest contribution proposed by Baraglia et al. [1].They proposed a WUM system called SUGGEST, that provide useful information to make easier the web user navigation and to optimize the web server performance. SUGGEST adopts a two levels architecture composed by an offline creation of historical knowledge and an online engine that understands user's behavior. As the requests arrive at this system module it incrementally updates a graph representation of the Web site based on the active user sessions and classifies the active session using a graph partitioning algorithm. Potential limitation of this architecture and other architectures might be quality of recommendations.

3 Architecture Figure of Online Predicting in WUM-Based Personalization System (OPWUMP)

The OPWUMP, shorting for the Online Predicting in WUM-based personalization system, is a Data Mining system that can be used for online predicting of users' request. According to different functions, the system can be partitioned into two main phases; offline phases and online phases. The architecture is shown in Fig 1.

3.1 Offline Phase

This phase consists of two major modules: Data pretreatment and Navigation Patterns Mining. In this phase we start with the primary Web-Log Preprocessing (Data pretreatment) to extract user navigation session from dataset and after that we will try to apply some algorithm to mining navigational patterns.

3.1.1 Data Pretreatment

Data pretreatment in a web usage mining model (Web-Log preprocessing) aims to reformat the original web logs to identify all web access sessions. The Web server usually registers all users' access activities of the website as Web server logs. Due to different server setting parameters, there are many types of web logs, but typically the log files share the same basic information, such as: client IP address, request time, requested URL, HTTP status code, referrer, etc.

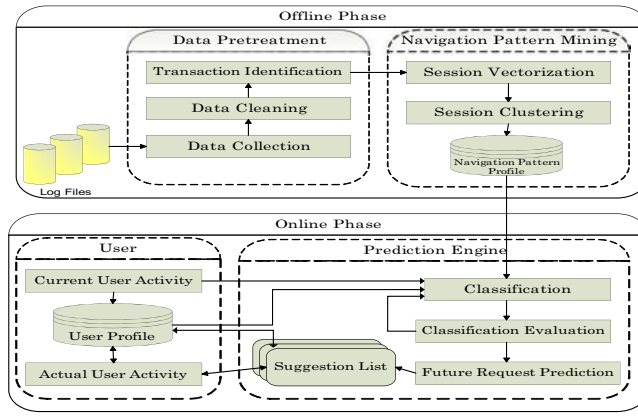


Fig. 1. The Architecture figure of OPWUMP

Generally, several pretreatment tasks need to be done before performing web mining algorithms on the Web server logs. For our work, these include data cleaning, user differentiation and session identification. These preprocessing tasks are the same for any web usage mining problem.

3.1.2 Navigation Pattern Mining

After the data pretreatment step, we will perform navigation pattern mining on the derived user access sessions. As an important operation of navigation pattern mining, clustering aims to group sessions into clusters based on their common properties. Since access sessions are the images of browsing activities of users, the representative user navigation patterns can be obtained by clustering them. These patterns will be further used to facilitate the user profiling process of our system.

For the session clustering we should assign a weight to web page visited in a session. The weight needs to be appropriately determined to capture a user's interest in a web page. For representing the interest degree of a web page to a user in the session, we can measure frequency and duration of a page in the session and sequence of accessed we pages.

In the session clustering module, standard clustering algorithm can partition user access sessions. The result of session clustering is used to represent the set of user navigation patterns. Given the transformation of user access sessions into a multidimensional space as vectors of web pages, standard clustering algorithms can partition this space into groups of sessions that are close to each other based on a distance measure. The results of sessions clustering will save in navigational pattern profile.

3.2 Online Phase

During the online phase, when a new request arrives at the server, the URL requested and the session to which the user belongs are identified, the underlying knowledge base is updated, and a list of suggestion is appended to the requested page.

The objectives of Prediction engine in this part of OPWUMP are to classify user navigation patterns and predict users' future requests. Classification module uses a

classification algorithm to classify current user activity .Each user session will be assigned a class label of pattern, so users' navigation activities can be clearly identified. We propose K-Nearest neighbor algorithm for classifying users' sessions. Next module of the architecture is evaluation of classification module, which aims to evaluate the results of classification based on some evaluation algorithm .If accuracy of classification does not meet satisfaction, classification module will be run again based on user profile and latest user activities. Otherwise prediction results achieve from future request prediction module. According to the prediction results, reasonable recommendations can be provided to the active session for better meeting of user's need.

The suggestion list module shows the prediction of users' future request. The users can choose a suggestion from suggestions list or maybe they continue their activities. Anyway each user action saves into user profile. This user profile will be used for improving accuracy of classification.

4 Conclusion and Future Work

In this paper, the architecture is proposed to classify user navigation pattern and Online predicting users' future request by mining of web server logs. This architecture will be used in the Web Usage Mining System named as OPWUMP. In this architecture, a prediction engine that works in online phase predicts next user request by interacting user profile and classification module. After classification part, accuracy of classification will be evaluated by evaluation part. If this accuracy doesn't satisfy the user, classification part will operate again based on latest user activity and user profile until the needed accuracy is satisfied.

This architecture can improve accuracy of classification and prediction in such systems. For the future, we would perfect the architecture and let it serves for actual users to the best of its abilities.

References

1. Baraglia, R., Silvestri, F.: Dynamic Personalization of Web Sites Without User Intervention. Communication of the ACM 50(20), 63–67 (2007)
2. Frias-Martinez, E., Karamcheti, V.: Reduction of user perceived latency for a dynamic and personalized site using web-mining techniques. In: WebKDD (2003)
3. Liu, R., Keselj, V.: Combined mining of Web server logs and web contents for classifying user navigation patterns and predicting users' future requests. Data & Knowledge Engineering, 304–330 (2007)
4. Mobasher, B., Cooley, R., Srivastava, J.: Automatic personalization based on web usage mining. Communications of the ACM 43(8), 142–151 (2000)
5. Yan, W.T., Jacobsen, M., Garcia-Molina, H., Umeshwar: From user access patterns to dynamic hypertext linking. In: Fifth International World Wide Web Conference (1996)

Artificial Intelligent Controller for a DC Motor

Hadi Delavari , Abolzafl Ranjbar Noiey, and Sara Minagar

Faculty of Electrical and Computer Engineering, University of Mazandaran, Iran
hdelavary@gmail.com, a.ranjbar@nit.ac.ir, sminagar@nit.ac.ir

Abstract. The Speed and position control of DC motors is addressed in this paper. An optimal intelligent control scheme is proposed for the system. Preliminary a PID controller is designed using Genetic Algorithms (GA). The proposed controller is implemented by using optimal integral state feedback control with GA and Kalman filter. In the proposed scheme, performance depends on choosing weighting matrices Q and R in the cost function, and accordingly GA is used to find these proper weighting matrices. In order to reduce the control performance degradation due to system parameters variation, a Kalman filter is gained. The performance of the proposed technique (ISF) is compared with PID controller. Computer simulation validates the effectiveness of the proposed scheme even in presence of uncertainties.

Keywords: DC motor, PID control, optimal control, Genetic Algorithm (GA).

1 Introduction

The DC machines have widely been used for high performance industrial applications. In spite of the variety in applications, speed and position control are the major task. An adaptive fuzzy PI sliding mode control is proposed in [1]. A fuzzy logic controller for position control of dc servo motor is given in [2]. Here an Optimal Intelligent Integral State Feedback Controller is proposed. At first a PID controller is designed, and then an intelligent optimal controller is applied.

2 Controller Design

The parameters of the PID controller obtained using a GA based optimization technique used in [3], have shown in Table 1. The parameters of an actual DC servomotor have listed in [4]. Although this PID controller has good performance, it is not so robust during the system parameters deviation, disturbance and noise [3]. Hence first the integral controller for the DC motor is designed and then a state feedback together with a Kalman filter is used. Assume the state space realization by:

$$\dot{x}(t) = Ax(t) + Bu(t) \quad , \qquad y(t) = Cx(t) + Du(t) \tag{1}$$

And $q(t)$ is assumed by the following definition

$$\dot{q}(t) = r - y(t) = r - Cx(t) - Du(t) \tag{2}$$

Then Kalman filter is used to improve the system performance.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 842–846, 2008.

Table 1. PID Parameter obtained by GA

PID Parameter	KP	Ki	Kd
position control	0.50142	0.1200	0.33011
Speed control	0.49663	1.0910	0.005998

In the case, some states are not available; Kalman filter as an optimal observer is used, to obtain estimation $\hat{x}$ of the real state variable x, this estimation can be used as a substitution for x, when required.

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + w \\ y(t) &= Cx(t) + v \end{aligned} \tag{3}$$

Where w is input noise and v is measurement. Then the estimation error will be:

$$e(t) = x(t) - \hat{x}(t) \tag{4}$$

The standard approach is used to estimate the state, employing the following closed loop observer structure:

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L[y(t) - C\hat{x}(t)] \tag{5}$$

$$\sum = E\left\{ e\, e^T \right\} \tag{6}$$

Kalman gain can be obtained by solving the following Algebraic Riccati Equation:

$$A\sum A^T - \sum - A\sum C^T (C\sum C^T + R_n)^{-1} C\sum A^T + Q_n = 0 \tag{7}$$

And then, the steady state Kalman gain is:

$$L = \sum C^T (C\sum C^T + R_n)^{-1} \tag{8}$$

Therefore, the state space model using (1), (2) and (5) can be written as:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{\hat{x}}(t) \\ \dot{q}(t) \end{bmatrix} = \begin{bmatrix} A & 0 & 0 \\ LC & A-LC & 0 \\ -C & 0 & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ \hat{x}(t) \\ q(t) \end{bmatrix} + \begin{bmatrix} B \\ B \\ -D \end{bmatrix} u + \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix} r \qquad y(t) = \begin{bmatrix} C & 0 & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ \hat{x}(t) \\ q(t) \end{bmatrix} + [D]u \tag{9}$$

Equation (8) defines the gain of state observer. And $u(t)$ is defined by:

$$u = \begin{bmatrix} -k_1 & -k_2 & -k_3 \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \\ q \end{bmatrix} + r \tag{10}$$

Therefore the state space model (8), (9) using (10) can be written as:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{\hat{x}}(t) \\ \dot{q}(t) \end{bmatrix} = \begin{bmatrix} A - BK_1 & -BK_2 & -BK_3 \\ LC - BK_1 & A - LC - BK_2 & -BK_3 \\ -C + DK_1 & DK_2 & DK_3 \end{bmatrix} \begin{bmatrix} x(t) \\ \hat{x}(t) \\ q(t) \end{bmatrix} + \begin{bmatrix} B \\ B \\ -D \end{bmatrix} r \tag{11}$$

$$u = \begin{bmatrix} C - Dk_1 & -Dk_2 & -Dk_3 \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \\ q \end{bmatrix} + [D]r \tag{12}$$

It must be noted that the optimal value for Q, R in the cost function is zero. This is a meaningless case because the tracking performance can not be observed. To cope with this problem a penalty term to the cost function is added [3]:

$$\int ((y(t)-r(t))^T (\mathbf{M}+\mathbf{Q})(y(t)-r(t))+u(t)^T \mathbf{R}\, u(t))dt \tag{13}$$

Where M is a positive semi definite matrix. In this paper optimal value for weighting matrices Q and R will be estimated by GA such a manner to reduce the settling time and overshoot as shown in Table 2. For simplicity, Q and R in the cost function are assumed diagonal. Step response for position and speed controller has been shown in Fig. 2. The output system response in the presence of noise and 20% parameter variation for the speed and the position controller can be seen in Fig. 3. Simulation results show the good performance, in presence of disturbance, noise and parameters variation in the plant. Applying Kalman Filter is shown in Table 2 signifying more robustness in the performance index.

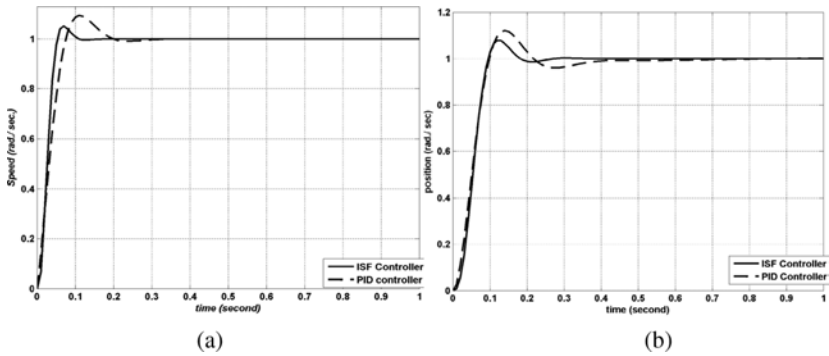
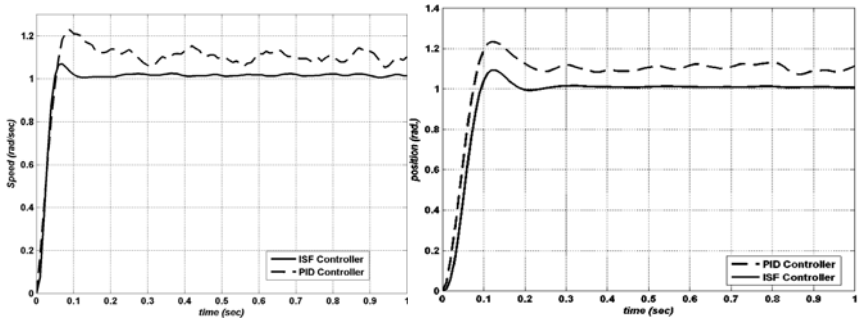


Fig. 2. Step response for: (a) Speed controller (b) position controller

Table 2. Speed controller and Position controller performance

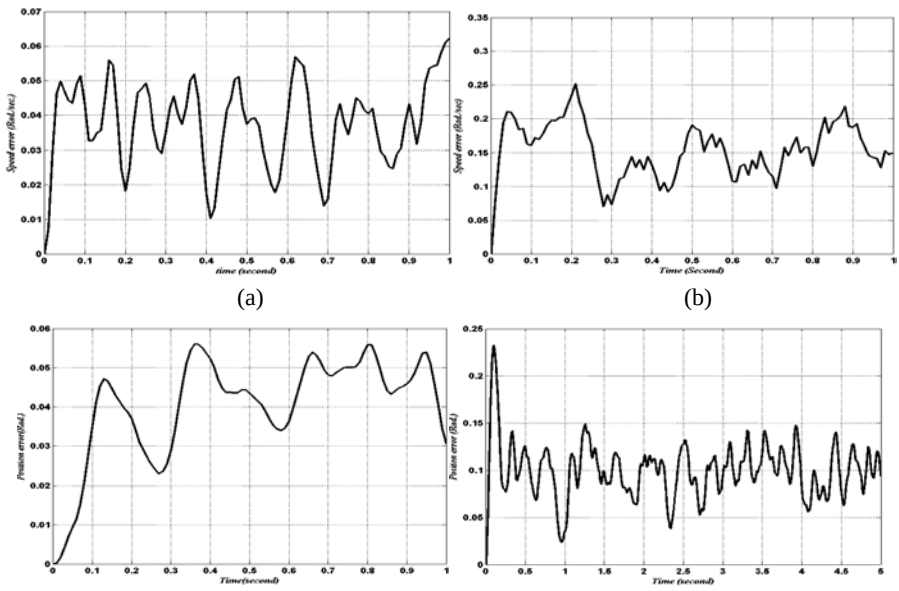
Time index	Speed controller performance		Position controller performance	
	PID controller	ISFcontroller	PID controller	ISFcontroller
settling time	0.176 (sec)	0.0878 (sec)	0.349 (sec)	0. 167 (sec)
overshoot	9.47%	5. 016%	12.2%	7. 88%

In Fig.4 the variations between the DC motor outputs with applying noise and without noise has shown. It is clear, DC motor outputs with ISF controller for both speed and position control has smaller variation than PID controllers, this means that ISF controller is more robust than PID controller. The simulation results reveal that using second controller to the speed control application, have shorter settling time, and smaller over shoot amplitude.



(a) (b)

Fig. 3. Step response in the presence of noise & 20% parameter variation for: (a) speed controller (b) position controller Control



(a) (b)

(c) (d)

Fig. 4. Output variation with noise and without noise: (a) for ISF in Speed control (b) for PID in Speed control (c) for ISF in Position control (d) for PID in Position

3 Conclusion

Design of a DC motor speed control system using both artificial intelligent optimal control and PID control has been presented in this paper. The problem of parameter uncertainties have been coped with using a Kalman filter.

The optimal value for weighting matrices Q and R, obtained by GA, achieving better performance in terms of the system characteristics i.e. settling time, overshoot

amplitude. According to the simulation results the latter has the better performance encountering the noise and disturbance and also the parameter variation in the DC motor plant. This proposed scheme can be used for every linear system.

References

1. Shahnazi, R., Shanechi, H., Pariz, N.: Position Control of Induction and DC Servomotors: A Novel Adaptive Fuzzy PI Sliding Mode Control. In: Power Engineering Society General Meeting, pp. 1–9. IEEE, Los Alamitos (2006)
2. Khongkoom, N., Kanchanathep, A., Nopnakeepong, S., Tamthong, S., Tunyasrirut, S.: Control of the Position DC Servo Motor by Fuzzy Logic. In: TENCON 2000 Proceedings, Kuala Lumpur, Malaysia, vol. 3, pp. 354–357 (2000)
3. Delavari, H., Alizadeh, G., Sharifian, M.B.B.: Optimal Integral State Feedback controller for a DC motor. In: ICEM 2006, Chania, Crete Island, Greece (2006)
4. Chew, M.Y., Menozzi, A., Holcomb, F.: On the comparison of the performance of emerging and conventional control techniques for DC motor velocity and position control. IEEE, Power Electronics and Motion Control 2, 1008–1013 (1992)

A Multi-Gb/s Parallel String Matching Engine for Intrusion Detection Systems

Vahid Rahmanzadeh[1] and Mohammad Bagher Ghaznavi-Ghoushchi[2]

[1] EE. Dept. School of Engineering, Tarbiat Modarres University, Tehran, Iran
[2] EE. Dept. School of Engineering, Shahed University, Tehran, Iran
rahmanzadeh@modares.ac.ir, ghaznavi@shahed.ac.ir

Abstract. This paper describes a Finite State Machine (FSM) approach on string matching for Intrusion Detection Systems (IDS). Search patterns are sliced into multiple interleaved substrings and feed into parallel FSMs. The final match results from combining the outputs of parallel individual FSMs. The proposed engine is primarily designed for ASCII codes and extended to support (16-bit) Unicode. The designed engine with 4-byte input words can reach search rates of over 30 Gb/s.

Keywords: String matching, Intrusion Detection Systems, Bit-splitting, Finite State Machine (FSM).

1 Introduction

String matching algorithms are crucial in text processing toolsets, data mining and especially in intrusion detection systems. In network intrusion detection systems, high throughput string matching engines are required to search in network traffic without degrading the speed of data transferring in network. Software string matching algorithms can reach the throughput of 250MHz at most [5]. In hardware solutions by using the natural properties of hardware such as parallelism, the Gb/s throughputs can be achieved [1].

Our work is based on two recent string matching techniques. First, Bit-Splitting for implementation of Finite State Machines [2] and second, parallel searching in multi-byte input words [1]. In our architecture, input sequence is partitioned to subsequences and each subsequence searched by one match module. In each clock cycle outputs of match modules are combined in an efficient way to determine whether a string pattern has been matched across all match modules or not.

The rest of this paper is laid out as follow. In section 2, the hierarchical structure of proposed string matching engine is described. Section 3 is about preprocessing algorithm. In section 4, simulation results are presented and conclusion is in section 5.

2 Architecture

Our architecture at the highest level is a full match machine. Full machine is composed of string matching engines that are structurally similar. All string patterns that

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 847 – 851, 2008.

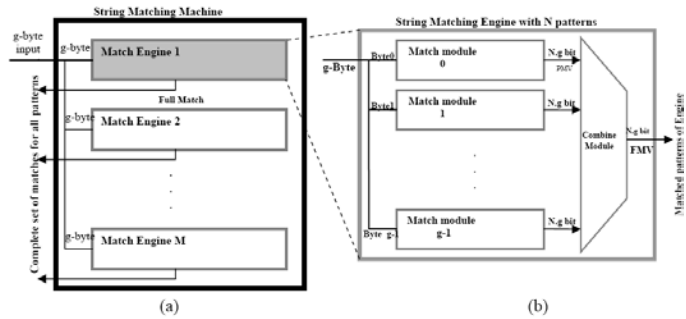


Fig. 1. Part (a) is a full string matching machine and part (b) is details of match engine

machine search them, classify to separated groups and each group is searched by one string matching engine. In each clock cycle, a g-byte wide input word is entered to machine and is sent to all parallel engines, where g is the length of input word in bytes and the number of match modules in a match engine.

Fig.1 (a) shows the full string matching machine, the part (b) of Fig.1 shows a match engine that is composed of g parallel match modules and a combine module.

Each match engine is composed of g-parallel match modules and a combine module. The set of g-byte input words are entering to engine each clock cycle. Bytes of input word distributed between g-match modules consecutively. Distribution is done based on index of each byte in input word. In this schema, input stream that is incoming to engine continuously, divided to g-individual subsequences in a g-interleaved manner.

For consistency, each string pattern also sliced to g-substrings in a way similar to input partitioning. It is supposed that a string pattern sliced to g-substrings from pat(0) to pat(g-1). At the first step for pattern splitting, pattern is divided to consecutive g-byte parts. Then by arranging the first byte of pattern parts consecutively, pat(0) is produced, pat(1) produced by second byte and pat(g-1) produced by g-th byte of pattern parts.

Substrings are entered to all match modules. Each match module is an implementation of Aho-Corasick string matching algorithm that searches all substrings of engine in its input subsequence and report matched substrings in output vector. Output vector of g-match modules is entered to combine module. In match modules Aho-Corasick algorithm implemented is based on bit-splitting technique by multiple parallel binary FSMs. Fig.2 shows structure of implemented match modules.

For the case of matching a string pattern, all substrings from pat(0) to pat(g-1) must match consecutively by match modules. Appropriate orders of substrings that are matched by match modules are checked in combine module. In our design, combine module is implemented by 'AND' and 'OR' gates to reduce delay of combine module. After implementation, structure of combine module is fixed and rule updating is done on the base of configuration of combine module of match engines.

Using ram-based technique for FSM implementation, allows non-interrupting rule update of match modules. Rules of each engine and total machine can be updated without stopping the operation of match machine. This rule updating process can complete in the order of seconds while FPGA based methods generally require days, to recompile rules. If rules are added one at a time, each match engine will take less than a total of g seconds to update [2].

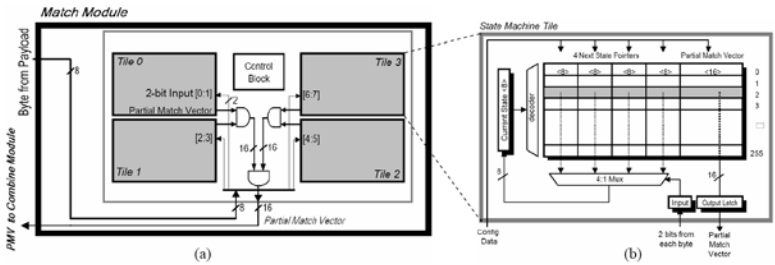


Fig. 2. Part (a) is the structure of match module and part (b) is a Tile implemented by table[ref.2]

3 Preprocessing Algorithm

In this section system software and data preprocessing algorithms implemented by C++ language are described.

At first step, in each match engine, strings sliced to g substrings in g-way interleaved manner. An Aho-Corasick FSM named D is built from substrings and each FSM is sliced apart into a new set of four FSMs, 2-bit groups of each byte of substrings are extracted to construct own Binary State Machine (BSM**)**. The alphabet elements of this FSM are {0, 1, 2, 3}. Binary FSMs of B0, B1, B2, B3 are built by splitting D in 2-bit groups, by the following steps.

Splitting starts from the initial state of D and all next states from root node in D-FSM are traversed. The output edges of each state are partitioned to four groups; those that are set to 0, 1, 2 or 3. These edges go to four new states in B. The process is repeated for other states of D to complete construction of B_i forward tree. Forward tree is the primary structure of FSM that in each state only success edges inserted. In next stage, failure edges are added to forward tree and matched patterns are added in each state. This part of algorithm is based on [2].

4 Simulation Results

Match modules are large FSMs implemented by bit-splitting into 4-tiny FSMs. Each FSM for 1-byte input words can achieve worst case throughput of 10-Gb/s for ASIC implementation [2], and 1.6-Gb/s for FPGA implementation on virtex 4fx100 [4]. By parallelism, total time of searching divided in two parts. One part is parallelizable time T_p that divided between parallel modules and reduced by factor g. Another part is serial time T_s that after parallelism being fixed or may be increased. Total serial time is sum of the original serial time of algorithm and inter-module communication that increased by parallelism. Speed-up formula for g stages parallelism is (1).

$$S(g) = \frac{T_s + T_p}{T_s + T_p / g} \tag{1}$$

As declared in (1), if delay of T_s is negligible as compare to T_p, speed-up can reach around g by g-stages parallelism.

In our proposed architecture, critical part of serial time is the delay of combine module. Combine module is implemented by g, g-input 'AND' gates and one g-input 'OR' gate. The implemented structure for combine module is really simpler than the structure of match modules. Therefore, the delay of combine module is much less than the delay of match module but yet, the delay of combine module vitiates the number of parallel match modules. By using pipelining, combining module works during the next clock cycle. It adds one clock cycle latency in design but throughput becomes linear on the number of match modules in each match engine.

In FPGA implementation, by using 4-parallel match modules, we achieved the throughput of 6.75Gb/s on virtex5. With this structure, throughput of over 30Gb/s for ASIC implementation can be achieved.

5 Conclusion

A parallel string matching for searching in multi-byte input words is developed. Input string patterns are divided to substrings in an interleaved way. Each substring is feed into all parallel FSMs of engine. The parallel FSMs search in separated input subsequences for substrings. The results of parallel FSMs are combined with a simple logic and matched patterns are determined.

Specially, this paper makes the following research contributions.

- Our structure search in multi-byte input words by parallel matching blocks in an interleaved way. The throughput of over 6.5 Gb/s on FPGA and over 30Gb/s for ASIC implementation can be achieved by 4 matching blocks in parallel.
- Our algorithm simply can be extended for processing larger input words in each clock cycle. On the other hand, Throughput can be increased linearly by using more parallel match modules.
- Our searching method and architecture can be easily extended for processing non-ASCII strings and languages in 16-bit Unicode.
- Our efficient architecture allows the rules to be updated without interrupting when string matching machine is working. The updating process can complete in the order of seconds.

References

1. Tripp, G.: A parallel String Matching engine for use in high speed network intrusion detection. J. computer virol. (2006)
2. Tan, L., Sherwood, T.: A high Throughput string matching architecture for intrusion detection and prevention. In: Proceedings of the 32nd International Symposium on Computer Architecture, ISCA 2005, pp. 112–122 (June 2005)
3. Aho, A.V., Corasick, M.J.: Efficient string matching: An aid to bibliographic search. Communications of the ACM 18(6), 333–340 (1975)
4. Jung, H.J., Backer, Z.K., Prasannaa, V.K.: Performance of FPGA Implementation of Bit-Split Architecture for Intrusion Detection Systems. IEEE, Los Alamitos (2006)

5. Aldwairi, M., Conte, M., Franzon, P.: Configurable string matching hardware for speedup up intrusion detection. In: Workshop on Architectural Support for Security and Anti-virus (WASSA) Held in Cooperation with ASPLOS XI (October 2004)
6. Attig, M., Lockwood, J.W.: SIFT: snort intrusion filter for TCP. In: Proceedings of IEEE symposium on high performance interconnects (Hot Interconnects-13), Stanford, California (2005)
7. Baker, Z.K., Prasanna, V.K.: A methodology for synthesis of efficient intrusion detection systems on FPGAs. In: Proceedings of IEEE symposium on field-programmable custom computing machines FCCM 2004, Napa, California (2004)
8. Sidhu, R., Prasanna, V.K.: Fast regular expression matching using FPGAs. In: Proceedings of the 9th international IEEE symposium on FPGAs for custom computingmachines, FCCM 2001, Rohnert Park, California, USA (2001)
9. Rahmanzadeh, V.: Generalized Approaches on 1D/2D FSPR-FSM design and Implementation. Ms. Thesis, Tarbiat modarres University, 1386, Tehran, Iran (2007)

A Modified Version of Sugeno-Yasukawa Modeler

Amir H. Hadad[1], Tom Gedeon[1], Saeed Shabazi[2], and Saeed Bahrami[3]

[1] Department of Computer Science and Information Technology,
Australian National University, Canberra, Australia
{Amir.Hadad,Tom.Gedeon}@anu.edu.au
[2] NICTA Victoria Laboratory,
Department of Computer Science and Software Engineering,
University of Melbourne
shahbazi@csse.unimelb.edu.au
[3] Department of Computer Science and Information Technology,
Abhar Azad University, Tehran, Iran
saeed.bahrami@aau.edu.ir

Abstract. One of the most significant steps in fuzzy modeling of a complex system is Structure Identification. Efficient structure identification requires good approximation of the effective input data. Misclassification of effective input data can significantly degrade the efficiency of the inference of the fuzzy model. In this paper we present a modification to the Sugeno-Yasukawa modeler [1] to improve structure identification by increasing the accuracy of effective input data detection. We improved Sugeno-Yasukawa Modeler by modifying the algorithm in two ways. Firstly, we used a new Trapezoid Approximation method based on [2] to improve estimation of membership functions. Secondly we change the modeling process of modeling. There exist some intermediate models in the Sugeno-Yasukawa modeling process, a combination of which will result in the final fuzzy model of the system. In the original modeling process, parameter identification is only done for the final fuzzy model. By doing the parameter identification for the intermediate fuzzy models, we have improved the accuracy of these intermediate models. The RC (Regularly Criterion) error has been reduced for intermediate fuzzy models and the MSE decreased without using the new Trapezoid Approximation method. By using the new trapezoid method, the RC value for the intermediate models and MSE for the final model improved even more. This accuracy increase, result in a better detection of effective input data among input data records of a system.

Keywords: Fuzzy Logic, Fuzzy Modeling, Trapezoid Estimation, Structure Identification, Parameter Identification.

1 Introduction

A common real world problem is to find the effective input parameters for each new environment on the basis of required outputs and create a relationship between these input/output parameters to be able to react appropriately in different situations. After

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 852–856, 2008.

creating these relationships, the rules and the way input/output parameters related to each other should be adjusted. The whole process is called modeling or identifying a model [3], and can be divided into: structure identification, parameter identification. Structure identification is of two types. The first is to find the effective parameters is called structure identification type I (some would call that feature selection [9])

The fuzzy model was firstly introduced by Zadeh ([1], [4]) and is a good choice to model a nearly unknown environment, as humans do. In this paper we have introduced a modified version of the Sugeno-Yasukawa modeling process.

In section two of this paper the modified version of Sugeno-Yasukawa modeling process will be explained. In the third section, we present the modified version of Trapezoid Approximation. In the simulation results section we will compare different methods from different perspectives and the result of the modifications in the original modeling process and the benefits and drawbacks of using this modified version of modeling process will be presented.

2 Modified Sugeno-Yasukawa Fuzzy Modeling Method

Our main modification is in "structure identification type I" phase. In the original process there is only one parameter identification phase for the final detected parameters. But in our method in each of the phases in detecting effective parameters we run the parameter adjustment phase for the membership function of the input and output parameter(s). Afterwards the RC[1] criterion is calculated on the basis the formula which is presented in [6].

By applying this algorithm on the intermediate models, their membership function parameters would be adjusted. This results in lower RC values in both effective and non-effective input data intermediate models.

RC values have decreased in comparison with the RC values in [1] by using the modified modeling method for the following sample function:

$$y = (1 + x_1^{-2} + x_2^{-1.5})^2, \quad 1 \le x_1, x_2 \le 5. \tag{1}$$

3 Creating Fuzzy Membership Functions for Sparse Data

In our method we modified the algorithm which was presented in [2]. The data of each cluster for creating the fuzzy membership functions of the original algorithm is selected based on the following steps:

1. $MaxU = Max\{U_1(x), U_2(x), ..., U_n(x)\}$
2. $\forall x \in M \text{ if } U_i(x) = MaxU \text{ then } x \in cluster_i$

Where:

n: Number of clusters,
U_i: Membership function of cluster *i*,
$cluster_i$: it represents the *i'th* cluster dataset,
M: Dataset.

[1] Regulatory Criterion.

In cases where the dataset is sparse, the number of data which belongs to each of the clusters in this method can be restricted and it is not possible to create the fuzzy membership functions by using the trapezoid algorithm which is presented in [2]. We created a new algorithm for these cases. We changed the data categorization part by using the following selection method:

$$\forall x \in M \text{ if } U_i(x) > boundary \text{ then } x \in cluster_i$$

Where:

U_i: Membership function of cluster i,

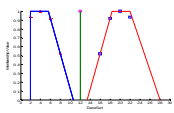
$cluster_i$: it represents the i'th cluster dataset,

boundary: it is a constant number less than or equal to 0.1

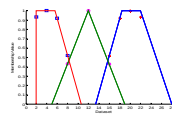
M: Dataset.

In this new method there is no need the calculate *MaxU*.

After the cluster data selection, we use the Trapezoidal Approximation based on [2].



(a) Old method trapezoidal estimation result, q = 3



(b) Our method trapezoidal estimation result, q = 3

Fig. 1. (a) Old method trapezoidal estimation result (b) Our method trapezoidal estimation result (q=3 in both (a) and (b)).

It is clear that without changing each cluster data selection method it is not possible to apply the trapezoidal algorithm on data.

4 Simulation Results

For the simulation, part we generate sample datasets for formula (1) 5 times. For each of the sample dataset generations, we generate 3 input variables one hundred times randomly and calculate the output based on input variables 1 and 2. We applied three structure identification algorithms on these datasets:

1. Original modeling method with original Trapezoid Approximation method: SY_{SY}
2. Our modeling method with Tikk's Trapezoid Approximation method: HA_{TI},
3. Our modeling method with our modified version of Tikk's Trapezoid Approximation method: HA_{HA}

To make it clearer how we improved the structure identification phase, we have divided the intermediate fuzzy models into two groups:

Group I: Models which only include effective input parameters
Group II: Models which include both effective and non-effective input parameters

We did a comparison between the average improvements in RC values of these groups. The average RC improvement value for the HA_{TI} method, group I is equal to 51.74%, while it is equal to -6.33% for group II. On the other hand, average RC improvement value for HA_{HA} method, group I is equal to 47.93%, while it is equal to

3.87% for group II (Table 1). We can conclude that the RC improvement for non-effective parameters in the modeling process is either negative or a small positive value. As a result, the detection of the effective parameters will be improved and the likelihood of detecting non-effective parameters as effective will be decreased. We calculated the RC improvement based on the following formula:

$$RC\ improvemen\ t(\%) = \frac{RC_{old} - RC_{new}}{RC_{old}} * 100 \tag{2}$$

We also calculated the final models, MSE shares based on the following formula:

$$MSE\ Share(Method_i) = \frac{MSE\ (Method_i)}{MSE(Method_i) + MSE(Method_j) + MSE(Method_k)} * 100 \tag{3}$$

Since we ran the simulation five times, the results presented in Table 2, is the averages of MSE shares for each of the methods.

We are doing extra calculation to decrease MSE of the final model and the drawback will be processing time increase for HA_{HA} and HA_{TI} method. Time taken for each of the methods for 5 runs is presented in Table 3, as we can see; the time for the SY_{SY} method is the lowest.

Table 1. Comparison of group I intermediate fuzzy models and group II intermediate fuzzy models RC improvement

	HA_{HA}	HA_{TI}
Group I Improvement	47.93%	51.74%
Group II Improvement	3.87%	-6.33%

Table 2. Average of MSE shares of final models for 5 runs of the algorithm for random data

	HA_{HA}	HA_{TI}	SY_{SY}
MSE Share	16.3%	33.8%	49.9%

Table 3. Average of time shares of final models for 5 runs of the algorithm for random data

	HA_{HA}	HA_{TI}	SY_{SY}
MSE Share	51.2%	42.4%	6.4%

5 Conclusion

We modified the Trapezoid Approximation presented in [2] and created broader membership functions. This slightly increased the RC values of the intermediate models in the HA_{HA} method in comparison to the HA_{TI} method. However, after applying parameter identification on the final model the MSE was lower in the HA_{HA}

method. Since membership functions are broader the chance of finding a more optimum membership function will be increased.

We have also boosted the process of structure identification strongly by doing the parameter identification process for intermediate fuzzy models. In this way, we have decreased the RC error of intermediate models for intermediate models. What makes our modified process more accurate in detecting effective input parameters from non-effective input parameters is that the RC values reduction in models of group I is more than that of group II.

The HA_{HA} method has the following advantages in comparison with the SY_{SY} method. Firstly, it is very useful when number of probably effective input parameters is large. Secondly, it is useful for sparse data for trapezoidal approximation.

Acknowledgments. Amir Hadad is funded by a grant form the Australian Research Council (ARC), Saeed Shahbazi is funded by National ICT Australia (NICTA).

References

1. Fukuyama, Y., Sugeno, M.: A new method of choosing the number of clusters for fuzzy c-means method. In: Proc. 5th Fuzzy system Symposium, pp. 247–250 (1989)
2. Tikk, D., Biró, G., Gedeon, T.D., Kóczy, L.T., Yang, J.D.: Improvements and Critique on Sugeno's and Yasukawa's Qualitative Modeling. IEEE Transaction on fuzzy systems 10(5) (October 2002)
3. Sugeno, M., Yasukawa, T.: A Fuzzy-logic-based Approach to Qualitative Modeling. Fuzzy Systems, IEEE Transactions on Fuzzy Systems 1(1), 7–31 (1993)
4. Zadeh, L.A.: Towards a theory of fuzzy systems. In: Kalman, R.E., DeClaris, R.N. (eds.) Aspects of Network and System Theory, pp. 469–490. Holt, Rinehart and Winston, New York (1971)
5. Zadeh, L.A.: Fuzzy sets and information granularity. In: Gupta, M., Ragade, R., Yager, R. (eds.) Advances in Fuzzy Set Theory and Application, pp. 3–18. North Holland, Amsterdam (1979)
6. Ihara, J.: Group method of data handling towards a modeling of complex systems-IV. Systems and Control (in Japanese) 24, 158–168 (1980)
7. Fukuyama, Y., Sugeno, M.: A new method of choosing the number of clusters for fuzzy c-means method. In: Proc. 5th Fuzzy System Symposium (in Japanese), pp. 247–250 (1989)
8. Chua, L.O., Yang, L.: Cellular neural networks: theory. IEEE Transactions on Circuits and Systems 35(10), 1257–1272 (1988)
9. Langley, P.: Selection of relevant features in machine learning. In: Proceedings of the AAAI Fall Symposium on Relevance. AAAI Press, Menlo Park (1994)

Directing the Search in the Fast Forward Planning

Seyed Ali Akramifar and Gholamreza Ghassem-Sani

Computer Engineering Department, Sharif University of Technology,
Tehran, Iran
akrami@ce.sharif.edu, sani@sharif.edu

Abstract. In this paper, we introduce DiFF, a novel extension of the Fast Forward (FF) planning system. FF is a domain independent planner that employs a forward heuristic search. Its search strategy is an enforced form of hill climbing. In order to move to a more promising state, FF evaluates successor states without any particular order. In this paper, we introduce a new form of the enforced hill climbing, which we call directed enforced hill climbing, to enhance the efficiency of Fast Forward planning. This strategy evaluates successor states in the order recommended by an adaptive heuristic function. Our experimental results in several planning domains show a significant improvement in the efficiency of the Fast Forward planning.

Keywords: AI Planning, heuristic search planning, fast forward planning system.

1 Introduction

In recent years, heuristic search planners highly outperform traditional planning systems. Heuristic search planners have been mostly based on a heuristic search in the space of world states. The first successful domain-independent planning system using this approach was HSP [1]. Other researchers adopted HSP's search method and applied various modifications, leading to the development of planners such as GRT [2], MIPS [3], AltAlt [4], and Fast Forward planning system (FF) [5]. Based on FF, some other powerful planners, such as Metric-FF [6], Conformant-FF [7], Fast Downward [8], and YAHSP [9], have also been developed. All these planners have been based on some heuristic functions. In most cases, the heuristic function approximates the distance between the current state and the goal state, based on a relaxed plan in which all effects that make the state variables false are ignored. The estimated value of a state corresponds to the size of the shortest relaxed plan.

This paper presents yet another new heuristic search planner based on FF, named Directed FF (DiFF). DiFF employs a double heuristic search strategy to perform more efficient in the planning.

The Enforced form of Hill Climbing (EHC) was introduced in the FF planning system [6] as a novel search strategy that combines the hill climbing and the breadth first search. The main challenge of a forward heuristic search planner is of course the proper selection of a node to develop next. Since state evaluation is costly, FF searches for the first better next state (if there exists any) rather than the best successor state. On the other hand, DiFF employs an auxiliary heuristic function, which can be used to propose more

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 857–861, 2008.

promising successor states, enabling the planner to choose more valuable states. We call this directed enforced hill climbing (DiEHC).

We have tested and compared the results of planning by FF and DiFF in some STRIPS planning domains. The experimental results show that in many domains, DiFF explores much fewer states than those explored by FF, and thus it is more efficient than the latter.

2 DiFF Planning System

DiFF planning system, which employs DiEHC, is an extension of FF planning system. Both of these planners use a heuristic function *h*, to evaluate search states, toward finding a solution. Heuristic function *h* estimates a distance between the current and the goal states. On the other hand, the main difference between FF and DiFF is in that FF uses the enforced hill-climbing as its main search strategy, whereas DiFF employs what we call directed enforced hill climbing.

The value of a state corresponds to how close it is to the goal state. This is too hard to be accurately determined, but it can be approximated. One popular approximating method is to solve a relaxed problem. Relaxation is achieved by ignoring the negative effects of actions.

2.1 Directed Enforced Hill Climbing

Directed enforced hill-climbing, is an extension of the enforced hill climbing augmented with an auxiliary heuristic function *O*. In any intermediate state s_c of a planning problem Π, DiEHC evaluates successor states in the order proposed by *O*. This heuristic function directs EHC through a potentially suitable path. Other steps are exactly the same as EHC.

Therefore, the main differentiating component of DiEHC is its ordering function *O*. This function should impose a minimum overhead. One way to minimize the overhead is to use, as much as possible, the past information that directed the planner to its current state. This information is highly dependent on the heuristic function *h*. accordingly, the ordering function *O* should be mainly based on function *h*.

2.2 State Priority Heuristic Function

The concept presented in this paper is general in that different ordering heuristic functions can be employed. It is important, however, that the ordering function *O* to be efficient. Here, we use an ordering function that is based on a negative credit assignment approach. In other words, the function assigns negative weights to those actions that produce unacceptable states. If the previous states generated by an action, *a*, were all unacceptable, new states generated by *a* might be unacceptable, too.

3 Results and Discussion

3.1 Planners, Benchmarks, and Objectives

In this section, we compare DiFF with FF in several classical planning domains. Our benchmark domains included: the classical Logistics and Blocks world domains, the

Mprime domain created for the 1st IPC, the FreeCell domain created for the 2nd IPC, Rovers, Depots, and the Satellite domains created for the 3rd IPC. Logistics and Blocks world problems are from the 2nd IPC, and FreeCell ones belong to the 3rd IPC, and Satellite problems was for the 4th IPC. We have divided these domains into two groups: The deadlock full domains, and The deadlock free domains.

All running times for DiFF and FF have been measured on a Pentium(R) 4 CPU running at 2.40 GHz, with 256 MB of main memory.

3.2 Deadlock Full Domains

We have categorized a number of classical planning domains to be deadlock full domains in that the enforced hill climbing may get stuck in some dead-ends, while trying to solve problems from such domains. In this respect, we tested FF and DiFF on 3 deadlock full domains: Depots, FreeCell, and the Mprime domains.

The Depots Domain. In this domain, DiFF was more efficient than FF, as it is shown in Figure 1-a. DiFF was more efficient than FF in almost all tested depots problems: up to 5.5 times faster in problem 19.

The FreeCell Domain. We classify the 20 evaluated problems in 3 groups: C1 that involves those problems that are solved by the main search strategy of FF and DiFF, C2 includes those problems that both EHC and DiEHC ordinarily failed to solve and had to resort to a BFS, and C3 comprises those problems that either EHC or DiEHC failed to solve, As Figure 1-b shows, DiFF was more efficient than FF in tackling problems of group C1. In C3, the planner that used its main hill-climbing strategy was more efficient.

The Mprime Domain. As Figure 1-c shows, DiFF was more efficient than FF in some problems and FF was more efficient than DiFF in solving some others).

3.3 Deadlock Free Domains

The deadlock free domains are those for which both of these enforced hill climbing search strategies are complete. We tested FF and DiFF on 4 deadlock free domains: The Logistics, the Rovers, the Satellite, and the Blocks world.

The Logistics Domain. In this domain, DiFF was much more efficient than FF: up to 33 times faster in problem 16 (see figure 1-d).

The Rovers Domain. In this domain, DiFF and FF produced plans with nearly the same lengths, but DiFF was more efficient, especially in larger problems (figure 1-e).

The Satellite Domain. In this domain, DiFF was again more efficient than FF, especially in larger problems (see figure 1-f).

The Blocks world. Figure 1-g show the planning time of only 27 Blocks world problems. DiFF and FF did have nearly similar results on other test problems.

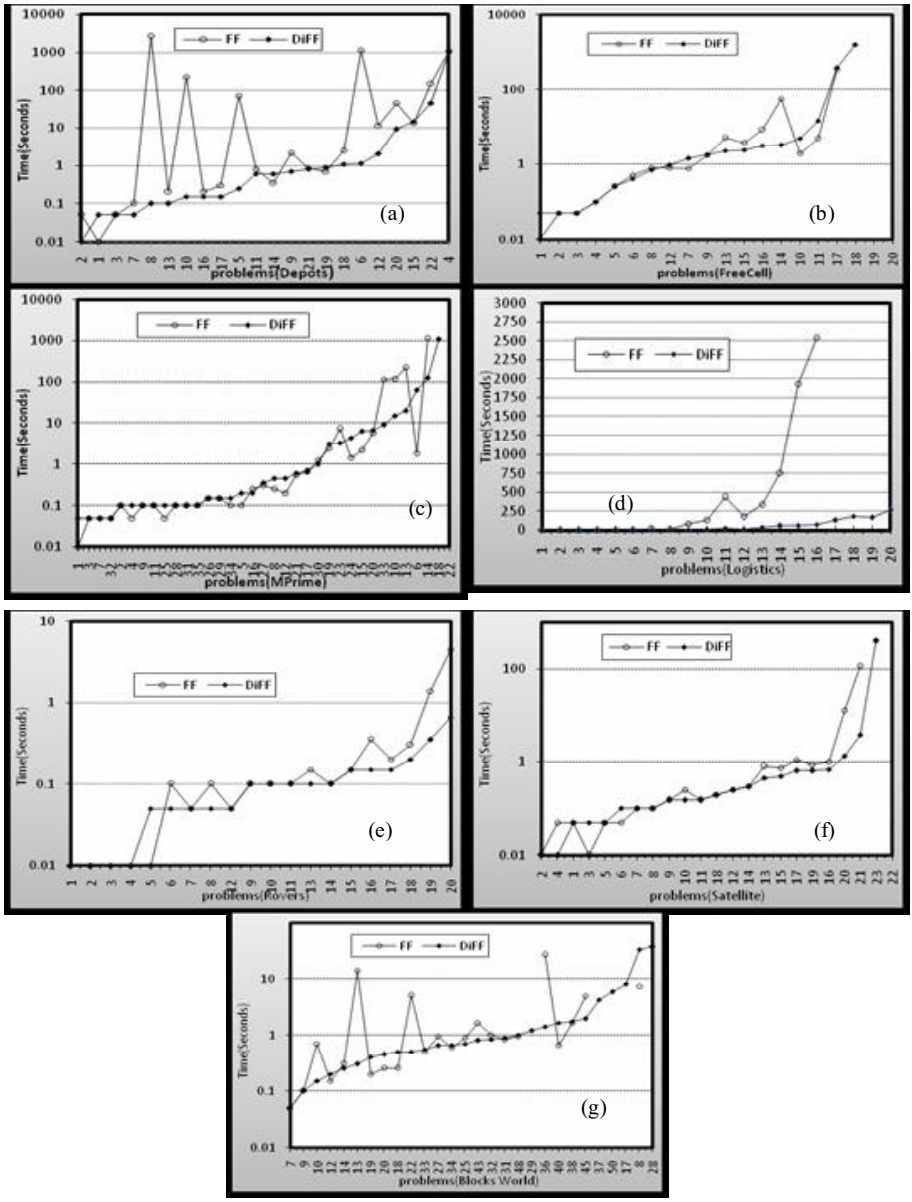


Fig. 1. Results of FF and DiFF in different domains

4 Conclusion and Future Works

Enforced hill climbing is the main search strategy used by the fast forward planner (FF). To evaluate the next states, FF examines candidate next states without any

particular order. We have implemented DiFF, which uses an improved search strategy. DiFF evaluates successor states in the order determined by an extra heuristic function. We called this strategy the directed enforced hill climbing. Ordering heuristic function used in DiFF monitors the planning process and learns from failures. The experimental results in several classical planning domains demonstrated that DiEHC examines fewer states than EHC, and thus is more efficient. However, in some planning domains, DiFF's performance is currently inferior to that of FF.

Using an extra ordering function to direct the search is a general idea. In other words, we can devise other heuristic functions to this goal. We are currently working on a different heuristic function that will use relaxed plans of the previous states to order successor states. This is a statistical function based on the actions included in those extracted relaxed plans. It is also interesting to see how we can apply the directing idea to other powerful heuristic search planners. The similarity between search strategies of these planners can help us to this aim. For this purpose, we are exploring the idea in other settings in particular in best first or A* style search, by trying more than one variant of the penalties and comparing their behaviors.

References

1. Bonet, B., Geffner, H.: HSP: Heuristic Search Planner. In: AIPS 1998 Planning Competition, Pittsburgh (1998)
2. Refanidis, I., Vlahavas, I.: The GRT planning system: Backward heuristic construction in forward state space planning. Journal of Artificial Intelligence Research 15, 115–161 (2001)
3. Edelkamp, S., Helmert, M.: MIPS: The model checking integrated planning system. AI Magazine 22, 67–71 (2001)
4. Nguyen, X., Kambhampati, S.: Extracting effective and admissible state space heuristics from planning-graph. In: AAAI 2002 (2002)
5. Hoffmann, J.: The Metric-FF planning system: Translating ignoring delete lists to numeric state variables. Journal of Artificial Intelligence 20, 291–341 (2003)
6. Brafman, R., Hoffmann, J.: Conformant planning via heuristic forward search: A new approach. In: 14th International Conference on Automated Planning and Scheduling (ICAPS 2004), Whistler, Canada (2004)
7. Helmert, M.: Fast Downward: Making use of causal dependencies in the problem representation. In: IPC 2004 (2004)
8. Vidal, V.: A Lookahead Strategy for Heuristic Search Planning. In: Fourteenth International Conference on Automated Planning and Scheduling ICAPS 2004, pp. 150–159 (2004)
9. Fikes, R.E., Nilsson, N.J.: STRIPS: A new approach to the application of theorem proving to problem solving. Artificial Intelligence 2, 189–208 (1971)

A Dynamic Mandatory Access Control Model*

Jafar Haadi Jafarian, Morteza Amini, and Rasool Jalili

Computer Engineering Department
Sharif University of Technology
Tehran, Iran
{jafarian@ce.,m_amini@ce.,jalili@}sharif.edu

Abstract. Mandatory access control has traditionally been employed as a robust security mechanism in critical environments like military ones. As computing technology becomes more pervasive and mobile services are deployed, applications will need flexible access control mechanisms. Aggregating mandatory models with context-awareness would provide us with essential means to define dynamic policies needed in critical environments. In this paper, we introduce a dynamic context-aware mandatory access control model which enables us to specify dynamic confidentiality and integrity policies using contextual constraints.

Keywords: Mandatory Access Control, Dynamicity, Context-Awareness, Confidentiality, Integrity.

1 Introduction

Mandatory access control is a means of restricting access to objects based on the sensitivity (as represented by a label) of the information contained in the objects and the formal authorization (i.e., clearance) of subjects to access information of such sensitivity. Mandatory access control has traditionally been employed as a robust security mechanism in critical environments like military ones. Due to high heterogeneity and dynamicity, new computing environments such as pervasive environments require more flexible authorization policies which are mainly dependant on context; e.g. an operation can only take place in a specified time interval [1]. Aggregating mandatory models with context-awareness would provide us with essential means to define dynamic policies needed in new pervasive critical environments. In other words, incorporating context-awareness into mandatory models would enormously enhance their expressiveness as well as dynamicity.

In this respect, Ray et al. [2] proposed a location-based mandatory access control model which extends Bell-LaPadula model with the notion of location. In particular, every location is associated with a confidentiality level and Bell-LaPadula no read-up and no write-down properties are extended by taking confidentiality levels of locations into consideration. Nonetheless, context-awareness has never been applied to mandatory access control models. In other words, although location is considered as a

* This research is partially supported by Iran Telecommunication Research Center (ITRC).

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 862–866, 2008.

fundamental contextual value, Ray et al.'s model can not be considered as a context-aware mandatory access control model.

In this paper, we introduce a dynamic context-aware mandatory access control model which firstly preserves the confidentiality and integrity of information and secondly enables us to define dynamic access control policies required in pervasive critical environments.

2 A Dynamic Mandatory Access Control Model

The model we present in this paper is a dynamic context-aware mandatory access control model which is preserving both confidentiality and integrity while utilizing contextual information to enhance expressiveness and dynamicity of traditional mandatory models. In particular, a subject's access to an object can be contingent on contextual values as well as confidentiality and integrity axioms.

The model can be formally defined as a ten-tuple:

$\langle EntitySet, RepOf, ConfLvl, IntegLvl, \lambda, \omega, ContextPredicateSet, ContextSet, OperationSet, OperatorDefiner \rangle$

in which *EntitySet* is the set of all entities in the system and is composed of four sets: *UserSet*, *SubjectSet*, *ObjectSet* and *EnvironmentSet*. *UserSet*, *SubjectSet* and *ObjectSet* include all users, subjects, and objects of the system respectively. *EnvironmentSet* has only one member called *environment*.

- *RepOf: SubjectSet* $\rightarrow$ *UserSet* determines for each subject the user on behalf of whom the subject is acting.
- *ConfLvl* is an ordered set of confidentiality levels. For instance in Bell-LaPadula confidentiality model [3], *ConfLvl* can be defined as $\langle TS, S, C, U \rangle$.
- *IntegLvl* is an ordered set of integrity levels. For instance in Biba integrity model [4], *IntegLvl* can be defined as $\langle C, VI, I \rangle$.
- $\lambda : (UserSet \cup SubjectSet \cup ObjectSet) \rightarrow ConfLvl$ is a mapping function which associates each user, subject, and object, with a confidentiality level.
- $\omega : (UserSet \cup SubjectSet \cup ObjectSet) \rightarrow IntegLvl$ is a mapping function which associates each user, subject, and object, with an integrity level.
- *ContextPredicateSet* is the set of current context predicates in the system (described in section 2.1). *ContextSet* is a set of context types (introduced in section 2.2). *OperationSet* is the set of all operations in the system (defined in section 2.3).
- *OperatorDefiner* function defines operators in $OperatorSet = \{\leq, \geq, <, >, \neq, \subseteq, \supseteq, \subset, \supset, \not\subset, =\}$ on contextual values as well as confidentiality/integrity levels. In other words, *OperatorDefiner* determines that for two arbitrary values *A* and *B* whether *A op B* is true or not. The *OperatorDefiner* is added to the system as an external module.

2.1 Context Predicate

Each *context predicate* is a predicate which represents a value for a contextual attribute. We define a context predicate $cp \in ContextPredicateSet$ as a 4-tuple:

$$cp = \langle en, ct, r, v \rangle$$

in which $En \in EntitySet$, $ct \in ContextSet$, $r \in RelatorSet_{ct}$ and $v \in ValueSet_{ct}$. For example, $\langle John, Location, Is, Classroom \rangle$ is a context predicate which gives information about the current location of subject *John*.

If $\langle e, x, r, v \rangle$ is a context predicate, $x[e][r]$ indicates the value assigned to entity *e* for context type *x* and relator *r*. In other words, $x[e][r] = v$. For instance if $\langle John, Location, Is, Classroom \rangle \in ContextPredicateSet$, then $Location[John][Is] = Classroom$. If such a context predicate does not exist in *ContextPredicateSet*, we assume that $x[e][r] = \perp$ (read as null).

2.2 Context Type

Informally, *context type* can be defined as a property related to every entity or a subset of existing entities in the system. In fact, context type represents a contextual attribute of the system; e.g. time or location of entities. Formally, a context type like $ct \in ContextSet$ can be defined as a binary tuple:

$$ct = \langle ValueSet_{ct}, RelatorSet_{ct} \rangle$$

More detail on each component of *ct* is given below.

Set of Admissible Values: $ValueSet_{ct}$
$ValueSet_{ct}$ denotes the set of values that can be assigned to variables of context type *ct*. Set representation can be used to determine members of $ValueSet_{ct}$. For instance, the value set of context type *Time* can be defined using set comprehension as follows:

$$ValueSet_{Time} = \{n \in N \mid 0 \le n \le 24\}$$

Set of Admissible Relators: $RelatorSet_{ct}$
$RelatorSet_{ct}$ represents the set of admissible relators for context type *ct*. For instance, for context type *Location*, $RelatorSet_{Location}$ can be defined as follows:

$$RelatorSet_{Location} = \{Is, Entering, Leaving\}$$

2.3 Operations

OperationSet is the set of operations defined in the system. Formally an operation $opr \in OperationSet$ is defined as $opr = \langle AccessRightSet_{opr}, Constraint_{opr} \rangle$.

More details on each component of the Operation *opr* are given below.

$AccessRightSet_{opr}$
The set of access rights in our model is comprised of *read (r)* and *write (w)*. In this model, every operation, based on what it carries out, includes a subset of these access rights; e.g. if it only does an observation of information and no alteration, it only includes *r*, and so on. $AccessRightSet_{opr}$ is a subset of the set {*r, w*} which denotes access rights of the operation.

$Constraint_{opr}$
Each operation owns a *constraint* which denotes the prerequisite conditions that must be satisfied before the operation is executed. For, $opr \in OperationSet$ this constraint is represented by $Constraint_{opr}$.

Use of variables *USR*, *SBJ* and *OBJ* allows us to define an operation constraint in a general way. When a subject as a representative of a user, wishes to execute an operation on an object, the variables are replaced by their current values and then the constraint is evaluated.

Operation constraints are logical expressions built of *condition blocks (CB)*. Each *CB* declares a conditional statement that can be evaluated using *OperatorDefiner* function. Formally a condition block is a triple $\langle Value_1, op, Value_2 \rangle$ in which $Value_1, Value_2 \in (ValueSet \cup IntegLvl \cup ConfLvl)$ and $op \in OperatorSet$ and is evaluated using the *OperatorDefiner* function in the following way:

$$OperatorDefiner(Value_1, op, Value_2)$$

For example, $\langle Location[SBJ][Is] = University \rangle$ denotes that the location of subject must be a subset of the university.

Operation constraint is constructed using the following unambiguous grammar:

$$Constraint \rightarrow Constraint \vee C1 \,|\, C1$$

$$C1 \rightarrow C1 \wedge C2 \,|\, C2$$

$$C2 \rightarrow (Constraint) \,|\, CB$$

Definition of operations finalizes the specification of our model. Next, we clarify the process through which the users' requests are authorized.

2.4 Authorization of Users' Requests

A subject's request to access an object is represented by an *action*. Formally, an action *A* is a triple $\langle s, o, opr \rangle$ in which $s \in Subject`$, $o \in Object$ and $opr \in Operation$. Since each subject is a representative of a user, the user of an action is determined by *RepOf(s)*.

Before an action is granted, its constraint must be evaluated. The constraint of an action such as *act* is denoted by $Constraint_{act}$ and is initially equal to the constraint of its operation; i.e. for an action $act = \langle s, o, opr \rangle$, initially $Constraint_{act} = Constraint_{opr}$.

The access rights set of an operation determines which mandatory policies must be considered before the operation is executed. In order to preserve confidentiality, we use Bell-LaPadula policy. Also, to preserve integrity, Biba strict integrity policy is being used in the model. In particular, for an action $act = \langle s, o, opr \rangle$, if $r \in AccessRightSet_{opr}$, Bell-LaPadula Simple-Security property and Biba Simple-Integrity property must be added to the constraint of that action. In addition, if $w \in AccessRightSet_{opr}$, Bell-LaPadula *-property and Biba Integrity *-property must be incorporated into it.

$$if\ r \in AccessRightSet_{opr} \quad Constraint_{act} = Constraint_{act} \wedge (\langle \lambda(SBJ), \geq, \lambda(OBJ) \rangle \wedge \langle \omega(OBJ), \geq, \omega(SBJ) \rangle)$$

$$if\ w \in AccessRightSet_{opr} \quad Constraint_{act} = Constraint_{act} \wedge (\langle \lambda(OBJ), \geq, \lambda(SBJ) \rangle \wedge \langle \omega(SBJ), \geq, \omega(OBJ) \rangle)$$

Before the constraint of an action is evaluated *USR*, *SBJ* and *OBJ* are replaced by their current values. For example for an action $act = \langle s, o, opr \rangle$ and *u* = *RepOf(s)*, *USR*, *SBJ* and *OBJ* are replaced with *u*, *s* and *o* respectively.

After the replacement is done the constraint is evaluated using the *OperatorDefiner* function and if the result of evaluation is true, the action will be granted.

3 Evaluation and Conclusions

Our model could be evaluated and compared with other mandatory models based on the following criteria: complexity of policy specification, support for context-awareness, expressiveness and security objective. Due to the space limitation we only consider complexity and expressiveness in this paper.

In respect of complexity, although specification of access control policies in our model is dynamic and flexible, it is more complicated in comparison with Bell-LaPadula and other traditional MAC models. However, the dynamicity which it offers, justifies its complexity in policy specification.

Furthermore, various mandatory policies can be specified by our model. Models such as Dion [5] and policies like Chinese Wall [6] can be easily expressed using our model. Therefore, the model is highly expressive.

In this paper, we explained the need for a dynamic mandatory access control model and presented a model which satisfies such a requirement. Our model utilizes context-awareness to enable specification of sophisticated and dynamic mandatory policies. In addition, various mandatory controls can be incorporated into our model. In this model, Bell-LaPadula, and Biba strict integrity policy are designated as built-in, and Chinese Wall and Dion policies can be appended to the model using context types.

One of the main advantages of the model besides its dynamicity is the ability of deploying different combinations of MAC policies simultaneously in a system. For instance, Bell-LaPadula, Biba, and Chinese Wall Policies can all be deployed at once.

References

1. Ray, Kumar, M.: Towards a location-based mandatory access control model. Computers & Security 25, 36–44 (2006)
2. Masoumzadeh, R., Amini, M., Jalili, R.: Context-Aware Provisional Access Control. In: Proceedings of the Second International Conference On Information Systems Security, Kolkata, India (2006)
3. Bell, D.E., LaPadula, L.J.: Secure Computer System: Unified Exposition and Multics Interpretation. MITRE Corporation (1976)
4. Biba, K.: Integrity Considerations for Secure Computer Systems, Bedford, MA (1977)
5. Dion, L.C.: A Complete Protection Model. In: Proceedings of the IEEE Symposium on Security and Privacy, Oakland, CA (1981)
6. Sandhu, R.S.: A Lattice Interpretation of the Chinese Wall Policy. In: Proceedings of the 15th NIST-NCSC Nat'l Computer Security Conference, Washington, D.C (1992)

Inferring Trust Using Relation Extraction in Heterogeneous Social Networks

Nima Haghpanah, Masoud Akhoondi, and Hassan Abolhassani

Computer Engineering Department
Sharif University of Technology
Tehran, Iran
{haghpanah,akhoondi}@ce.sharif.edu, abolhassani@sharif.edu

Abstract. People use trust to cope with uncertainty which is a result of the free will of others. Previous approaches for inferring trust have focused on homogeneous relationships and attempted to infer missing information based on existing information in a single relationship. In this paper we propose using methods of social network analysis to infer trust in a heterogeneous social network. We have extended the problem of relation extraction and allowed using any type of binary operator on matrixes, whereas previous work have focused on linear combination of base matrixes (the only allowed operator was summation of two matrixes). We present two genetic algorithms which use ordinary numerical and fuzzy logic operators and compare them on a real world dataset. We have verified our claim – ability to infer trust in a heterogeneous social network- using proposed methods on a web-based social network.

Keywords: Trust, Trust Model, Relation Extraction, Graph Mining.

1 Introduction

Trust is a concept which researchers have attempted to model in their implementations of security technologies [4,5]. Previous approaches for inferring trust do not use the interrelated information in heterogeneous social networks [2,3]. In this paper we propose using the methods from heterogeneous social network analysis to infer trust in such networks. We define a relation extraction problem, which was first proposed by [1] and has been applied to DBLP dataset to investigate community structure in research interests of authors of scholarly papers.

The process of combining multiple base relations to form a new relation based on previous knowledge about the target relation is called relation extraction. This previous knowledge can be provided in form of simple queries. This operation can be used extensively in social community websites, for friend suggestion, targeted marketing, and network prediction, to name but a few [1]. Previous work on relation extraction has focused on linear combination of base relations. In this paper we extend the problem to fuzzy logic operators and show that it can yield better results.

We illustrate the idea by an example. Consider a social network such as Orkut, Tickle, Friendster, in which people form communities based on their interests, culture, and beliefs. People may gather in a community based on their common interest in

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 867–870, 2008.

certain genre of music or literature, their nationality, their religion, or any other issue which is common between them. We view these communities as relationships between objects, here people. Two people are considered related in a relationship if they are (or are not) both members of the assigned community. Furthermore, each person assigns any other one a degree of trust, which may be unknown to us. Trust can also be seen as a relationship between people. We claim that this relationship can be inferred using a relation extraction technique given other relationships between people. We provide the algorithm a query, which is simply a number of people whose degree of trust between them is completely known to us. We then ask the algorithm to model complete relationship by combining other relationships. For example, such an algorithm may reveal that having the same nationality and educational level are highly related to trust, but having the same sexuality is totally unrelated.

Solutions have been proposed for relation extraction on heterogeneous social networks [1]. Our solutions outperform previously proposed solutions in their generality, scalability, extensibility, and interpretability. These solutions can handle different types of queries (single- and multiple-community), different number of base relations and query sizes. Furthermore, these solutions allow limiting the search space, which results in more interpretable solutions in cases where the user prefers to limit the number of combined relations to a constant. We define the constrained version of relation extraction problem and show that current algebraic methods are inapplicable in this case.

2 Trust in Heterogeneous Social Networks

Problem Definition. Cai et al. [1] formulate the relation extraction problem as follows:

"Given a set of objects and a set of relations which can be represented by a set of graphs $G_i(V, E_i)$, $i=1,\ldots,n$, where n is the number of relations, V is the set of nodes (objects), and E_i is the set of edges with respect to the i-th relation. The weights on the edges can be naturally defined according to the relation strength of two objects. We use M_i to denote the weight matrix associated with G_i, $i=1,\ldots,n$. Suppose there exists a hidden relation represented by a graph $G'(V,E')$, and M' denotes the weight matrix associated with G'. Given a set of labeled objects $X=[x_1,x_2,\ldots,x_m]$ and $Y=[y_1,y_2,\ldots,y_m]$ where y_j is the label of x_j (such labeled objects indicate partial information of the hidden relation G'), find a linear combination of the weight matrices which can give the best estimation of the hidden matrix M'."

This definition can be generalized to use any kind of operator:

Let O be any binary operator on $k\times k$ matrixes with indexes from -1 to 1. The combination of a number of base relations using this operators is another matrix $M = O_{j=1}^{n} a_j$. The generalized version therefore asks us to find the sequence of matrix coefficients $A=(a_1,a_2,\ldots a_n)$ which minimize the Frobenius norm between the difference of M and Q.

We also define the constrained version of relation extraction problem as follows:

Consider M_i, G_i, G', and M' as defined previously. The problem is to find up to k base matrixes whose linear combination is the best estimation of hidden matrix M'.

Algorithm 1. Genetic Algorithms are inspired by biological systems' improved fitness through evolution. A solution to a given problem is represented in the form of a string, called 'chromosome', consisting of a set of elements, called 'genes', that hold a set of values for the optimization variables.

In both of our algorithms, a solution is a sequence of size n, $S=(s_1,s_2,\ldots,s_n)$ in which the number at i-th position indicates the coefficient of matrix M_i. Each s_i can take values between 0 and 1. Each individual S in the algorithm therefore generates a solution $\sum_{i=1}^{n} s_i$. We limit search space by assigning a threshold on the number of nonzero coefficients in each solution. Every solution is only allowed to have up to k nonzero coefficients, where k is a predefined constant defined by constrained relation extraction problem (for the unconstrained problem, set $k=n$). We should define crossover and mutation operators in a way that they comply with this constraint.

Let M_S be the matrix associated with a solution S, $M_S = \sum_{i=1}^{n} s_i . M_i$. Fitness of such an individual would be the Frobenius norm of M_S-Q, where Q is the target matrix. That is, fitness(S)= $\sum_{i=1}^{n}\sum_{j=1}^{n}(m_{s_{ij}} - q_{ij})^2$.

Algorithm 2. Issues of representation, crossover and mutation operators are the same as algorithm 1. We only need to change the fitness function to use any new operator. We here choose to use fuzzy XOR operator. The fitness of each solution is computed as follows. For every solution $S=(s_1,s_2,\ldots,s_n)$, the matrix associated with it, M_S, is computed as follows. $M_S = XOR_{i=1}^{n} a_{i.M_i}$. We only need to define fuzzy XOR of two matrixes. We do this by defining the fuzzy XOR of two matrixes as bitwise XOR of the indexes of the two matrixes.

3 Experimental Results

The dataset which we used is part of the Trust Project at http://trust.mindswap.org/, and which is a network consisted of distributed data on the semantic web [2]. In this network, users rate the people they know according to their trust.

In the networks used for these results, users employed a 1-10 scale with discrete integer values. As a result, there is not a smooth distribution. Although one might expect the average trust rating to be 5, the middle value, this turns out not to be the case. The average trust rating is 7.25, with a standard deviation of 2.30 in the Trust Project network. In the network, there were 22,231 ratings assigned. The distribution has been scaled to show what proportion of the total number was comprised by each individual rating.

We selected a subset of 100 people and extracted their mutual trust in a form of a matrix. We used Orkut (www.orkut.com) to find communities which are most common between the selected people. We selected 10 most common communities and using these communities we formed 10 base relations. Two people are considered related in a base relation if either both of them or none of them are members of that community. We also selected 10 people from these 100 as our query to our algorithms; we gave trust ratings between these 10 people and also their relationships in

base relations. The algorithms then computed a 100*100 matrix which is their prediction of the trust relationship between each two people in our community. We used actual information about trust ratings between people to evaluate algorithms. The average error rate of our algorithms was 31% and 18%, respectively.

Fig 1. Shows the results of our algorithm on the dataset. The error rate is the average difference between the values computed by our algorithms and real values, which decreases as the number of base relations used increases. We changed the number of base relations used from 10 percent to 100 percent of total base relations to investigate the effect of increasing the number of base relations.

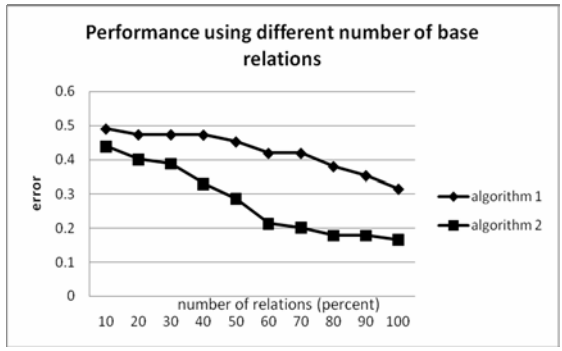


Fig. 1. Error rate using different number of base relations

References

1. Cai, D., Shao, Z., He, X., Yan, X., Han, J.: Community Mining from Multi-Relational Networks. In: Jorge, A.M., Torgo, L., Brazdil, P.B., Camacho, R., Gama, J. (eds.) PKDD 2005. LNCS (LNAI), vol. 3721. Springer, Heidelberg (2005)
2. Golbeck, J.: Computing and Applying Trust in Web-Based Social Networks. PhD thesis, University of Maryland, College Park (2005)
3. Golbeck, J., Hendler, J.: Filmtrust: Movie recommendations using trust in web-based social networks. In: Proc. of the Consumer Communications and Networking Conference (2006)
4. Mui, L., Mohtashemi, M., Halberstadt, A.: A computational model of trust and reputation. In: Proceedings of the 35th International Conference on System Science, pp. 280–287 (2002)
5. Sabater, J., Sierra, C.: Reputation and social network analysis in multi-agent systems. In: Proc. of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 475–482. ACM Press, New York (2002)

Sorting on OTIS-Networks

Ehsan Akhgari[1], Alireza Ziaie[1], and Mohammad Ghodsi[2,*]

[1] Computer Engineering Department, Sharif University of Technology, Tehran, Iran
[2] Computer Engineering Department, Sharif University of Technology, Tehran, Iran
School of Computer Science, Institute for Studies in Theoretical Physics and Mathematics, Tehran, Iran
{akhgari,ziai}@ce.sharif.edu, ghodsi@sharif.edu

Abstract. Because of their various desirable properties, OTIS networks are subject to active research in the field of parallel processing. In this paper, the OTIS network architecture is presented, emphasizing two important varieties, the OTIS-mesh and the OTIS-hypercube. Next, a randomized sorting algorithm presented in the literature for OTIS-mesh structures will be reviewed. Finally, a discussion of sorting on OTIS-hypercubes will be offered, and a suggested algorithm for sorting on these networks, with an MPI implementation will be introduced.

Keywords: OTIS Network, OTIS-Mesh, OTIS-Hypercube, Sort algorithms.

1 Introduction

Optical interconnection overcomes the known problems of electrical interconnection, such as the communication bottlenecks in high bit rates and the limitations in case of long distances. Hence, hybrid architectures using both optical and electrical communication links such as the "OTIS[1] Network" are under active research.

OTIS is an optoelectronic interconnected network developed for parallel processing systems. The main idea in OTIS Networks is using optical links alongside with electronic links. This idea was first introduced in [2]. In OTIS, processors are partitioned into groups (clusters). Intra-group connections are electronic, and the Inter-group connections are optical. For more information on OTIS networks, refer to [4].

The groups inside an OTIS network can be of any of the various parallel processing architectures, including meshes and hypercubes. The resulting architecture will be named based on the underlying group architecture – for example, OTIS-mesh and OTIS-hypercube are OTIS networks resulting from connecting mesh and hypercube structures, respectively. A comparison of the properties of OTIS-mesh and OTIS-hypercube architectures has been presented in [3].

2 OTIS Network Structure

The Intra-group connections (electronic connections) in OTIS networks vary based on the architecture selected for the groups, but the inter-group connections (optical

* This author's work was in part supported by a grant from IPM (N. CS2386-2-01.).
[1] Optical Transpose Interconnection System.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 871–875, 2008.

connections) are always formed in a well-known fashion. Suppose that each node is assigned a label in form of (g, p), where g is the group number, and p is the node number in each group. For all i and j, where $i \neq j$, the node (i, j) is connected to the node (j, i) using an optic link.

3 Sorting on OTIS-Hypercube Networks

An N^2 node OTIS-hypercube is built using N clusters, each being a $\lg N$ -cube. The hypercube nodes in each cluster are linked together by electronic links. Following the same convention as the OTIS-Mesh network for numbering the nodes, each node will be assigned a number in form of (g, p), where g is the group number, and p is the node number in each cluster. Using this naming convention, every (p, g) node in the network is connected to the (p, g) node by means of an optic link. A sample OTIS-hypercube structure appears in Fig. 1. This structure consists of eight 3-cube clusters, as well as the optical links between them.

We will proceed to show that an N^2 node OTIS-hypercube has less hardware cost than an N^2 node hypercube. An N^2 node hypercube has $N^2 \lg N$ edges, which are all electronic. A similar N^2 node OTIS-hypercube network, on the other hand, consists of N hypercubes, each with $\frac{1}{2} N(\lg N)$ edges, which makes a total of $\frac{1}{2} N^2(\lg N)$ electronic edges. In addition, there are $\frac{1}{2}(N^2 - N)$ pairs in the form of (i, j) where $0 \leq i, j < N$ and $i \neq j$. Therefore, there are $\frac{1}{2}(N^2 - N)$ optic edges as well in the OTIS-hypercube network. Therefore, the total number of edges in an N^2 OTIS-hypercube is $\frac{1}{2}(N^2(\lg N) + (N^2) - N)$.

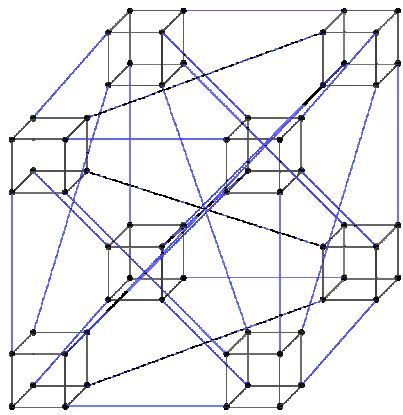


Fig. 1. A 64-node OTIS-hypercube network, where electronic and optic links are shown using black and blue links, respectively

In order for an N^2 node OTIS-hypercube network to have less edges than an N^2 node hypercube, the following inequality must hold.

$$2\left(N^2(\lg N)\right) > N^2(\lg N) + \left(N^2\right) - N \,. \tag{1}$$

Eq. 1 can be rewritten in form of $N(\lg N) > N - 1$, which holds for each $N > 1$. Because the special case $N = 1$ has only a single processor, and cannot be utilized for any parallel computations, this special case can be ignored. Therefore, we have shown than an N^2 node hypercube has more edges than an N^2 node OTIS-hypercube for all practically interesting networks. Furthermore, $\frac{1}{2}\left(N^2 - N\right)$ of the edges of such an OTIS-hypercube network are optical. So, one can safely conclude that the OTIS-hypercube network is less costly than a hypercube network with the same number of nodes.

Such a comparison only applies where one of the two networks can be simulated using the other one with a small slowdown. The following theorem establishes that an N^2 node OTIS-hypercube network can actually simulate an N^2 node hypercube.

Theorem 1. An N^2-node OTIS-Hypercube network can simulate an N^2-node hypercube with a slowdown factor of at most 3.

The proof of this theorem appears in [5]. The proof is explained informally here as well, because it is used in construction of the sorting algorithms on OTIS-hypercube networks. Suppose the node (g, p) in the OTIS-hypercube is labeled in form of $(g_1 \cdots g_n p_1 \cdots p_n)$, where $g_1 \cdots g_n$ is the binary representation of g, $p_1 \cdots p_n$ is the binary representation of p and $n = \lg N$.

For an OTIS-hypercube to be equivalent to a hypercube, every node in the network must be connected to all nodes which differs from it in exactly one bit position. If the difference bit position is in the lower half (i.e., in $p_1 \cdots p_n$) there is a direct electronic link between the nodes, because both are located in the same cluster. If the difference bit position is in the upper half, the link can be simulated in at most 3 steps. The first step is using an optic link to move to the node $(p_1 \cdots p_n g_1 \cdots g_n)$ (the g and p parts of the label are swapped.) After this step, the difference bit lies in the lower half of the label, so the transform can be made locally in the cluster. In the third step, another optic link is traversed to restore the order of g and p in the label.

This simulation can happen in 2 steps for the special cases where one of the source or destination addresses is in form of (i,i). If the source address is in that form, the first step of the above simulation is unnecessary (and not possible, since nodes with (i,i) addresses do not have any optic adjacent edges.) Likewise, if the destination address is in that form, the last step of the simulation would be redundant.

From Theorem 1, it can be concluded that any hypercube algorithm can be simulated on an OTIS-hypercube network with the constant slowdown of at most 3 (thus, asymptotically equivalent). This includes sorting algorithms on hypercubes as well. Therefore, these algorithms can also be implemented on OTIS-hypercubes, by substituting data movements along the higher dimension half of hypercube nodes with the 3-step simulation laid out above.

We will proceed to explain how one sample hypercube sorting algorithm can be adapted to OTIS-hypercube networks. The chosen algorithm is the Odd Even Merge Sort on a hypercube [1]. This algorithm works by recursively merging larger and larger sorted lists. In particular, to sort N items, it begins with splitting the items into N sublists of length 1. Then, pairs of unit-length lists are merged into $N/2$ sublists of length 2. The same process goes on so that in the last step of the recursion, 2 sublists of length $N/2$ are merged to form the final sorted sequence. The running time is $O(\lg^2 N)$, which doesn't change for the OTIS-hypercube adapted algorithm. The total number of steps in the original algorithm is $\lg^2 N + \lg N$.

The interesting part of the algorithm, as described in [1] is the merging part. The implementation of merging is described on a butterfly network. On a butterfly network, the merging step of the algorithm consists of two passes. During each step of the first pass, the location of the data items in the upper half of the butterfly does not change (the straight edges are used), while the data in the lower half moves towards the cross edges. The last step of the first pass is exceptional in that all of the straight edges are used (even in the lower half). The first pass effectively reverses the order of data items in the lower half. During the second pass, the data flows back in the butterfly network. At each step, if the adjacent pairs are out of order, they are switched using the cross edges, otherwise they are simply moved using the straight edges. The merge step of the algorithm takes $2\lg N$ steps.

Since this algorithm is normal, it can be run in the same running time on a hypercube. To simulate it on an OTIS-hypercube network, the following points must be considered.

- The last step of the first pass has no effect on a hypercube (it simply leaves the position of all of the hypercube edges unchanged). Therefore, it can be omitted in the OTIS-hypercube simulation.
- The first pass of the algorithm consists of $\lg N - 1$ steps. The first $\frac{1}{2}\lg N$ steps occur on the electronic edges, and therefore can be simulated with no slowdown. Each of the next $\frac{1}{2}\lg N - 1$ steps occur in 3 steps in an OTIS-hypercube network. Altogether, the first pass would take $2\lg N - 3$ steps.
- The second pass consists of $\lg N$ steps. The first $\frac{1}{2}\lg N$ steps can each be simulated in 3 steps, and the next $\frac{1}{2}\lg N$ can be directly simulated on an OTIS-hypercube using the electronic links. Therefore, the second pass would take $2\lg N$ steps.

Therefore, the total running time of the merge step would be equal to $4\lg N - 3$. So the total time for sorting N numbers will be

$$(4\lg 2 - 3) + \cdots + (4\lg N - 3) \quad = \quad 2\lg^2 N - \lg N . \tag{2}$$

The total number of steps in the original algorithm is $\lg^2 N + \lg N$. The simulated algorithm, as previously noted, has the same asymptotic running time ($O(\lg^2 N)$).

Specifically, the simulated algorithm takes at most $\lg^2 N - 2\lg N$ more steps to complete than the original algorithm.

4 Conclusion

This paper reviewed the important related work in the field of sorting algorithms for OTIS networks, and outlined one randomized sorting algorithm on OTIS-mesh architectures. In addition, a new sorting algorithm for OTIS-hypercube structures was suggested.

References

1. Leighton, F.T.: Introduction to Parallel Algorithms and Architectures: Arrays - Trees - Hypercubes. Morgan Kaufmann, San Francisco (1992)
2. Marsden, G., Marchand, P., Harvey, P., Esener, S.: Optical transpose interconnection system architectures. Optics Letters 18(13), 1083–1085 (1993)
3. Hashemi Najaf-abadi, H., Sarbazi-Azad, H.: An empirical comparison of OTIS-mesh and OTIS-hypercube multicomputer systems under deterministic routing. In: 19th IEEE International Parallel and Distributed Processing Symposium, p. 262a. IEEE Press, New York (2005)
4. Pijitrojana, W.: Optical transpose interconnection system architectures (2003)
5. Zane, F., Marchand, P., Paturi, R., Esener, S.: Scalable network architectures using the optical transpose interconnection system (OTIS). Journal of Parallel and Distributed Computing 60(5), 521–538 (2000)

A Novel Semi-supervised Clustering Algorithm for Finding Clusters of Arbitrary Shapes

Mahdieh Soleymani Baghshah and Saeed Bagheri Shouraki

Sharif University of Technology, Tehran, Iran
soleyman@ce.sharif.edu, bagheri-s@sharif.edu

Abstract. Recently, several algorithms have been introduced for enhancing clustering quality by using supervision in the form of constraints. These algorithms typically utilize the pair wise constraints to either modify the clustering objective function or to learn the clustering distance measure. Very few of these algorithms show the ability of discovering clusters of different shapes along with satisfying the provided constraints. In this paper, a novel semi-supervised clustering algorithm is introduced that uses the side information and finds clusters of arbitrary shapes. This algorithm uses a two-stage clustering approach satisfying the pair wise constraints. In the first stage, the data points are grouped into a relatively large number of fuzzy ellipsoidal sub-clusters. Then, in the second stage, connections between sub-clusters are established according to the pair wise constraints and the similarity of sub-clusters. Experimental results show the ability of the proposed algorithm for finding clusters of arbitrary shapes.

Keywords: Semi-supervised, clustering, pair wise constraints, sub-clusters.

1 Introduction

Until now, a large number of algorithms for discovering natural clusters in datasets have been introduced. In real-world problems, clusters can appear with different shapes, sizes, data sparseness, and degrees of separation [1]. In the last decade, many algorithms have been introduced for finding clusters of different shapes [2]. It seems that these unsupervised algorithms may not distinguish clusters of arbitrary shapes in different situations. Indeed, each of these algorithms has its own limits and yet no single algorithm is able to identify all sorts of cluster shapes and structures that are encountered in practice [1]. Recently, the problem of clustering with side information is receiving increasing attention [3]. Side information is valuable to the clustering problem, owing to the inherent arbitrariness in the notion of a cluster [4]. It can help to identify the cluster structure that is the most appropriate (according to the side information) among different possibilities of clustering. Although, many semi-supervised clustering algorithms [3-15] have been introduced in the last decade, these algorithms usually do not provide the ability to distinguish clusters of arbitrary shapes.

In this paper, a novel semi-supervised clustering algorithm is introduced that can find clusters of different shapes and sizes. This algorithm uses pair wise constraints as the most natural form of side information to find clusters. In the proposed algorithm,

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 876–879, 2008.

different clusters can be modeled using different number of prototypes. Indeed, in the first stage of this algorithm, a set of sub-clusters (prototypes) is found. Then, connections are established between these sub-clusters according to the constraints and a similarity measure between sub-clusters. Finally, the resulted graph is decomposed into the final clusters.

The rest of this paper is organized as follows: In Section two, our proposed semi-supervised clustering algorithm is presented. The results of applying this algorithm are reported in Section three. Finally, the last section presents conclusions.

2 Semi-supervised Clustering Algorithm

In this section, a two-stage semi-supervised clustering algorithm is introduced. Indeed, our clustering algorithm introduced in [2] is extended to a semi-supervised clustering algorithm. One of the most natural forms of side information in the clustering problems is a set of must-link or must-not-link constraints. In the first stage of the semi-supervised clustering algorithm, the data points are grouped into relatively large number of sub-clusters such that each sub-cluster must not contain any two data points appearing in a must-not-link constraint. Then, in the second stage, the pair wise constraints along with the similarity measure between sub-clusters are used to find final clusters as a set of connected components in the graph of sub-clusters. In the following subsections, these stages are illustrated.

2.1 Finding the Sub-clusters

In order to specify the sub-clusters in a dataset, the Gustafson-Kessel (GK) clustering algorithm is used. In this algorithm, a covariance matrix and a center vector are considered for each cluster. Thus, GK algorithm distinguishes clusters of ellipsoidal shapes. The resulted sub-clusters must not contain any pair of data points appearing as must-not-link constraint. Since the number of sub-clusters is often relatively large, this situation does not usually occur. However, if at least one of the must-not-link constraints is not satisfied, this stage of algorithm is repeated by setting the initial positions of some cluster centers to the data points appearing in must-not-link constraints. During the GK clustering algorithm, the position of these centers must not be changed. Thus, the data points appearing in must-not-link constraints are lied in different sub-clusters after the termination of the GK algorithm.

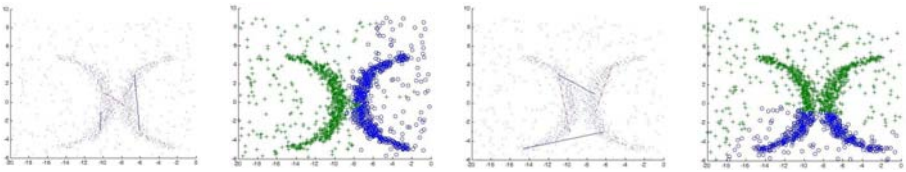
2.2 Forming Clusters Using Sub-clusters and Side Information

To form the final clusters, a graph of sub-clusters is created and the connected components of it are specified. Nodes of this graph are the sub-clusters and edges of it are specified according to the pair wise constraints and similarity of the nodes (sub-clusters). Since, the sub-clusters have been considered as elliptical shapes modeled using centers and covariance matrices, the Bhattacharya distance [2] can be used as a distance measure between two sub-clusters. In this stage, at first, for each data point appearing in a constraint, nearest sub-cluster to it is found and data points in the constraints are replaced by nearest sub-cluster to them. Indeed, must-link and must-not-link constraints on data points are changed to constraints on corresponding sub-clusters.

To find the clusters, a set of edges are created in the graph of sub-clusters such that the pair wise constraints are satisfied and the similarity between the connected sub-clusters (other than those connected because of the must-link constraints) should be as high as possible. Before finding the clusters, a list of edges (sub-cluster pairs) is created. Those pairs appearing in must-link constraints are put at above of this list. Then, the other pairs of sub-clusters are sorted according to the similarity of the sub-clusters (appearing in these pairs) in a descending order and appended to the edge list. In the next stage, a legal set of edges (containing must-link edges) that does not make contradiction with must-not-link constraints are found. Indeed, the connected components created by adding these edges to the graph must not contain any sub-cluster pair appearing in a must-not-link constraint. Also, the weights of the selected edges must be as small as possible. The backtracking technique has been used for this purpose.

3 Experimental Results

In this section, the result of applying our proposed semi-supervised clustering algorithm on a data set is presented. In Fig. 1(a) and Fig. 1(c), one dataset has been considered with two sets of constraints. The results of applying our proposed algorithm for clustering of these two cases have been shown in Fig. 1(b) and Fig. 1(d) respectively. The must-link and must-not-link constraints provided as side information have been displayed by filled and dashed lines respectively in this figure. The number of sub-clusters and clusters have been set to $S = 20$ and $C = 2$ respectively.



(a) Side info. 1. (b) Clustering result (a) (c) side info. 2. (d) Clustering result (c).

Fig. 4. The semi-supervised clustering result on Dataset1 according to side information 1

4 Conclusion

In this paper, we introduce a novel semi-supervised clustering algorithm using pair wise constraints to lead the search for finding the appropriate clusters. In our proposed algorithm, at first, the data points are grouped into relatively large number of elliptical sub-clusters. Then, the final clusters are formed by establishing connections between these sub-clusters using pair wise constraints and similarity of them. As opposed to the other existing semi-supervised clustering algorithm, the proposed algorithm can find clusters of different shapes, sizes, and densities while satisfying constraints.

References

1. Fred, A.L.N., Jain, A.K.: Combining multiple clusterings using evidence accumulation. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(6) (2005)
2. Soleymani Baghshah, M., Bagheri Shouraki, S.: A fuzzy clustering algorithm for finding arbitrary shaped clusters. In: Soleymani Baghshah, M., Bagheri Shouraki, S. (eds.) 6th ACS/IEEE International Conference on Computer Systems and Applications (to be published, 2008)
3. Basu, S.: Semi-supervised clustering: probabilistic models, algorithms and experiments. Ph.D. Thesis, University of Texas at Austin (2005)
4. Law, H.C.: Clustering, dimensionality reduction, and side information. Ph.D. Thesis, Michigan University (2006)
5. Zhengdong, L., Leen, T.: Semi-supervised learning with penalized probabilistic clustering. In: Neural Information Processing Systems, vol. 17 (2004)
6. Bansal, N., Blum, A., Chawla, S.: Correlation clustering. Machine Learning 56(1-3), 89–113 (2004)
7. Lange, T., Law, M.H., Jain, A.K., Buhmann, J.B.: Learning with constrained and unlabelled data. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 730–737 (2005)
8. Zhao, Q., Miller, D.J.: Mixture modeling with pair wise, instance-level class constraints. Neural Computation 17(11), 2482–2507 (2005)
9. Kulis, B., Basu, S., Dhillon, I., Mooney, R.: Semi-supervised graph clustering: a kernel approach. In: 22nd International Conference on Machine Learning, pp. 457–464 (2005)
10. Shental, N., Bar-Hillel, A., Hertz, T., Weinshall, D.: Computing Gaussian mixture models with EM using equivalence constraints. In: Neural Information Processing Systems, vol. 16, pp. 465–472 (2004)
11. Xing, E.P., Ng, A.Y., Jordan, M.I., Russell, S.: Distance metric learning, with application to clustering with side information. In: Neural Information Processing Systems, vol. 15, pp. 505–512. MIT Press, Cambridge (2003)
12. Bar-Hillel, A., Hertz, T., Shental, N., Weinshall, D.: Learning a mahalanobis metric from equivalence constraints. Journal of Machine Learning Research 6, 937–965 (2005)
13. Yeung, D.Y., Chang, H.: Extending the relevant component analysis algorithm for metric learning using both positive and negative equivalence constraints. Pattern Recognition 39, 1007–1010 (2006)
14. Yeung, D.Y., Chang, H.: Robust path-based spectral clustering. Pattern Recognition 41, 191–203 (2008)
15. Ceccarelli, M., Maratea, A.: Improving fuzzy clustering of biological data by metric learning with side information. International Journal of Approximate Reasoning (2007) doi:10.1016/j.ijar.2007.03.008

Improving Quality of Voice Conversion Systems

M. Farhid and M.A. Tinati

Tabriz University, Tabriz, Iran
morfid@gmail.com,
tinati@tabrizu.ac.ir

Abstract. New improvement scheme for voice conversion are proposed in this paper. We take Human factor cepstral coefficients (HFCC), a modification of MFCC that uses the known relationship between center frequency and critical bandwidth from human psychoacoustics to decouple filter bandwidth from filter spacing, as the basic feature. We propose U/V (Unvoiced/Voiced) decision rule such that two sets of codebooks are used to capture the difference between unvoiced and voiced segments of the source speaker. Moreover, we apply three schemes to refine the synthesized voice, including pitch refinement, energy equalization, and frame concatenation. The acceptable performance of the voice conversion system can be verified through ABX listening test and MOS grad**.**

Keywords: Voice conversion, HFCC, Vector quantization.

1 Introduction

The goal of a voice conversion system is to transform the speech spoken by a source speaker, such that a listener would perceive this speech as uttered by a target speaker. Voice conversion offers applications in a lot of fields, like Text-to-Speech adaptation, media entertainment (dubbing movie and karaoke), and even in the field of voice disguise. A general framework for voice conversion with basic building blocks is shown in Fig 1. The technique of converting voice quality from one speaker to another through vector quantization and spectral mapping has been investigated in [1]. In this method, vector quantization is used to partition the spectral feature space into subspaces. Each subspace is represented by its mean vector (centriod). Therefore, each speaker's individuality could be characterized by a set of centriods. The correspondence of centriods of the spectral spaces between two speakers defines the mapping function, which is used for conversion purposes. In contrast to vector quantization based methods, the classification of spectral data in the Gaussian mixture model (GMM) approach is probabilistic and is a continues function of spectral parameters, resulting in a “soft classification” of feature space[2]. This characteristic avoids the discontinuities which are inherent in the vector quantization models.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 880–883, 2008.

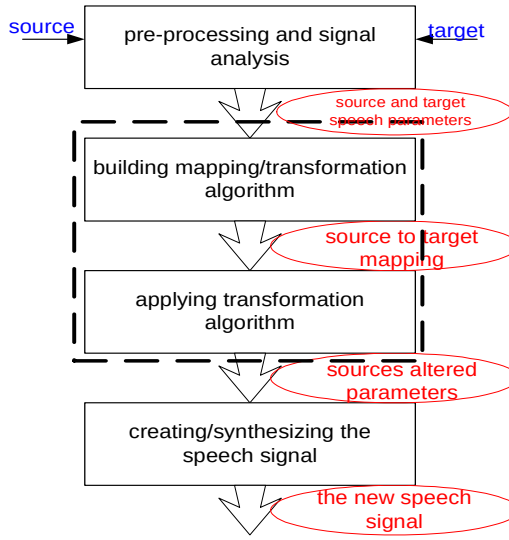


Fig. 1. Basic voice conversion block diagram

2 Proposed Algorithm

Our proposed method to refine the training and synthesizing procedures in voice conversion is as follows. We employ the human factor cepstral coefficients (HFCC) as the basic features. Before constructing the mapping codebooks, we classify speech frames into three categories: silence, unvoiced, and voiced. Codebooks for unvoiced and voiced frames are constructed separately such that the mapping between consonants and vowels can be described more precisely. Besides, we choose data grouping induced by alignment of dynamic time warping (DTW) to generate codebooks of the target speech. Moreover, in order to generate high quality voice, it is necessary to perform post-processing scheme to refine the synthesized speech, as described in the following sections.

2.1 Decision Rule for Unvoiced/Voiced Frames

Human's speech signals could be generally divided into two categories: unvoiced and voiced. The decision rule for U/V (unvoiced/voiced) classification is based on Zero Crossing Rate and Energy and Cross-correlation criteria [4].

2.2 Human Factor Cepstral Coefficient Filter Bank (Hfcc)

The Human Factor Cepstral Coefficients (HFCC), introduced by [3] represents the most recent update of the MFCC filter bank. HFCC, like MFCC, is not a perceptual model of the human auditory system but fairly a biologically inspired feature extraction algorithm. Assuming sampling frequency of 12500 Hz, [3] proposed the HFCC filter bank composed of 29 mel-warped equal height filters, which cover the frequency range [0, 6250] Hz. Since here a sampling frequency of 8000 Hz is assumed, only the first 24 filters, which cover the frequency range of [0, 4000] Hz, were kept.

2.3 Codebook Generation for the Source Speaker

The purpose of this step is to prepare the representative frames (prototypes or centers) of the source speaker for the subsequent alignment procedure between the source speaker (speaker A) and the target speaker (speaker B) via DTW. We use vector quantization to extract two codebooks for unvoiced and voiced frames of the source speaker. Figure2 shows the flowchart of U/V codebooks generation for the source speaker. we choose data grouping induced[4] by alignment of dynamic time warping (DTW) to generate codebooks of the target speech.

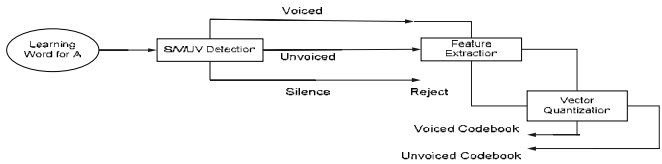


Fig. 2. U/V codebooks generation for the source speaker

Figure3 demonstrates the flowchart of the mapping codebook generation. Once the training procedure is accomplished, we can acquire the mapping codebook. Next, we adopt concatenation-based method to articulate each frame.

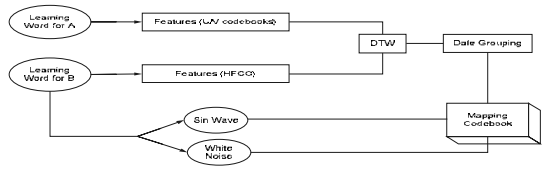


Fig. 3. Mapping codebook generation

3 Post Processing

3.1 Pitch Contour Modification

The pitch contour modification involves matching both the pitch mean value and range. The modified pitch, p_{mod}, is obtained by modifying the source speaker pitch by applying the following mapping function**:**

$$P_{\text{mod}} = A \log P_s + B \tag{1}$$

Where P_s is the source pitch period of the current frame and parameters A and B are defined in [5].

4 Experimental Results

To subjectively evaluate the performance of our system, two forced-choice (ABX) experiments and an MOS (mean opinion score) tests were performed. We assume ABX experiments to compare the improvement of the proposed method. The MOS experiment was carried out here to estimate the listening quality, using a 5-point scale: 1-bad, 2-poor, 3-fair, 4-good, and 5-excellent.

Table 1. Results of perceptual tests. (ABX test: Test question: "X is closer to A or to B?").

	Tests
ABX(MFCC)	89.0%
MOS(MFCC)	3.3
ABX(HFCC)	93.0%
MOS(HFCC)	3.7

References

1. Abe, M., Nakamura, S., Shikana, K., Kuwabara, H.: Voice Conversion though Vector Quantization. In: Proc. ICASSP, New York, USA, vol. 1, pp. 65–658 (1988)
2. Stylianou, Y., Cappe, O., Moulines, E.: Continuous probabilistic transform for voice conversion. IEEE Trans. Speech Audio Proc. 6, 131–142 (1998)
3. Skowronski, M.D.: Human Factor cepstral coefficient. Computational NeuroEngineering Laboratory University of Florida (2004)
4. Lin, C.-Y., Roger Jang, J.-S.: New Refinement Schemes For Voice Convertion. IEEE, Los Alamitos (2003)
5. Hassan, M.M.: A statistical mapping approach to voice conversion. Acoustics Soc. (December 2005)

Feature Selection SDA Method in Ensemble Nearest Neighbor Classifier

Fateme Alimardani, Reza Boostani, and Ebrahim Ansari

Department of Computer Science and Engineering, Shiraz University, Shiraz, Iran
alimardani@cse.shirazu.ac.ir, boostani@shirazu.ac.ir,
ansari@cse.shirazu.ac.ir

Abstract. The curse of dimensionality is still a big problem in the pattern recognition field. Feature extraction and feature selection have been presented as two general solutions for this problem. In this paper, a new approach based on combination of these methods has been proposed to classify different classes in large dimensional problems. Among the vast variety of search strategies, *Tabu Search (TB)* is chosen here as a core for feature selection. Filters & Wrappers are the two traditional types of objective functions for feature selection which both applied in this study. Following a feature extraction approach is considered. Subclass Discriminant Analysis (SDA), as a successful strategy in feature extraction has been employed. the result of each objective function on the standard UCI datasets are shown. Finally a revised nearest neighbor classifier has been used to classify the patterns in the new feature space on the UCI data sets and the results show the supremacy of our combinatorial approach in comparison with the traditional methods.

Keywords: Feature selection, Tabu search, Filter and Wrapper, Davies Bouldin, Subclass Discriminant Analysis (SDA). Ensemble Nearest Neighbor Classifier (ENNC).

1 Introduction

Many pattern recognition methods suffer from high dimensionality of data. In this way, a lot of researches have been performed to solve this problem . Feature selection and feature extraction are common solutions to solve curse of dimensionality. In this article, an investigation approach is proposed to use the advantages of both techniques. Here, a Backward Selection sequential search strategy has been applied for FSS. Feature selection algorithms commonly use one of two groups of objective functions: *Filters* and *Wrappers* [8]. Based on the previous experimental results, it is observed that each of them has some advantages and disadvantages [5]. Here, not only filter approach has been used but also one of the successful wrapper techniques has been employed to increase the performance. Among the vast variety of search strategies, TS is chosen for feature selection which is discussed in section 2. In practice, it is observed that in high dimensional problems, some features are noisy which reduce the classification accuracy. Thus, a two layers algorithm is proposed here to overcome this drawback. The first step selects a clean and non-noisy subset of features in order

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 884–887, 2008.

to remove the affect of noisy value on classification performance. The second step maps the data to a lower dimensional space by using an effective feature extraction algorithm (SDA). Discriminant Analysis (DA) [2, 3] and Linear Discriminant Analysis (LDA) [4] and several extension of them [3,7] have been successfully used as a feature extraction technique. Among the DA algorithms, SDA [1] approximate the distribution of each class with a mixture of Gaussians (GMM) [5] which gives a single formulation that can be used for most distributions. In section 3, SDA is introduced. In section 4, a new Ensemble Nearest Neighbor (ENN) classifier is proposed.

2 Tabu Search

The basic concept of Tabu Search (TS) as described by Glover [6, 1] is a meta-heuristic superimposed on another heuristic. A memory forces the search to explore the search space such that entrapment in local minima is avoided. The efficiency of Tabu search has been proven in many optimization problems.

Davies Bouldin criterion as a filter, and the classification error, as a wrapper are employed here. Both of the above criteria indexes should be minimized. Davies Bouldin index, gives a measure of clusters separability. The Davies-Bouldin index is defined as [9]:

$$DB(C) = \frac{1}{K}\sum_{i=1}^{K} \max\{\Delta(C_i) + \Delta(C_j) / \delta(C_i, C_j)\} \quad (1)$$

Where $\Delta(C_i)$ is the intra-cluster distance while $\delta(C_i, C_j)$ is the inter-cluster distance. Lower the index value (1), show the more separability of feature subsets.

3 Subclass Discriminant Analysis

It is well-known that LDA uses the between and within class scatter matrices. Earlier, we mentioned SDA divide classes into subclasses regarding to distinct clusters in the reduced space. S_B is defined as:

$$S_B = \frac{1}{n}\sum_{i=1}^{C-1}\sum_{j=1}^{H_i}\sum_{k=i+1}^{C}\sum_{l=1}^{H_k} p_{ij}\, p_{kl}(\mu_{ij} - \mu_{kl})(\mu_{ij} - \mu_{kl})^T \quad (2)$$

Where $p_{ij} = \frac{n_{ij}}{n}$ is the prior of the j^{th} subclass of class i .SDA with stability criteria is defined in [1] has been employed here.

4 Ensemble Nearest Neighbor

Common distance-based classification methods typically include naive Bayes, decision tree, k-Nearest Neighbor (k-NN), etc. In this article to compute the similarity of train and test sets, three kinds of distance measurements have been used which

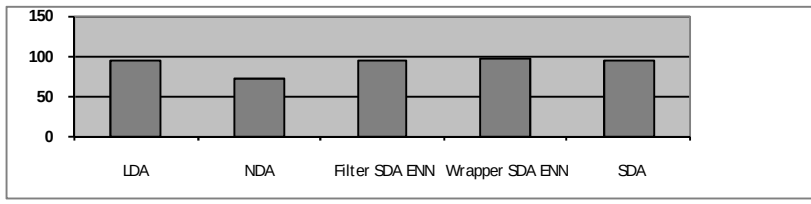
contain: 1 - Euclidean distance, 2 – Manhattan distance, 3 - Minkowski distance. In the next step a voting procedure is employed, and we assign the weight of each vote equals to one.

5 Experimental Result

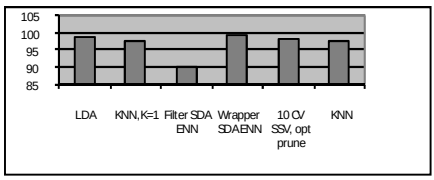
In order to assess the performance of the proposed method, we used six standard data-sets available from UCI ML repository (Table 1). The maximum accuracy on WDBC dataset is reported 94%, but our proposed method considerably improves the

Table 1. Resuls on datasets based on two type of objective function

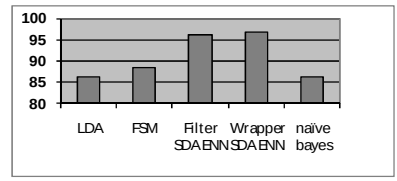
Dataset	Method	No. features from Tabu search	No.features from SDA stability	Test_error
WDBC	Classification-rate	28	15	2.9730
WDBC	Davies-Bouldin	25	21	3.8173
Glass	Davies-Bouldin	8	5	29.5900
Glass	Classification-rate	8	5	29.5900
Wine	Classification-rate	12	2	0.5263(1.6)
Wine	Davies-Bouldin	7	6	1.7403(2.3)
Pima	Classification-rate	7	6	29.8479
Pima	Davies-Bouldin	6	5	30.8489
Sonar	Classification-rate	54	45	17.723(2.7)
Sonar	Davies-Bouldin	52	9	16.4502(2.2)
Hepatitis	Davies-Bouldin	16	9	18.70
Hepatitis	Classification-rate	17	11	11.2132



(a)



(b)



(c)

Fig. 1. Average results of classification of the proposed method in comparison withsome other classifiers (a) on WBDC dataset , (b) on Wine dataset,(c) on Hepatitis

accuracy to 97.03%. Ten times 10-folds cross validation is applied on the datasets. Results of the proposed method are summarized in Table 1. The accuracy rate of Ensemble NN and some commonly used classifiers are shown in Fig. 1. signatures are acceptable.

6 Conclusion

In this paper, we defined a two-step feature reduction algorithm and proposed a new NN classifier to improve the classification performance. The important point is that, the developed algorithm is independent of the type of objective function. Our results show that the implementations of proposed method yield the higher classification rate on some data sets. Moreover, the algorithm can cope with the noisy features in the training data.

Acknowledgments. The author gratefully acknowledges E.Ashoor, E.Chitsaz, M. Taheri, for their help in doing the project.

References

1. Zhu, M., Martinez, A.M.: Subclass Discriminant Analysis. Transactions on Pattern Analysis and Machine Intelligence 28(8), 1274–1286 (2006)
2. Duda, R.O., Hart, P.E.: Pattern Classification and Scene Analysis, 1st edn. John Wiley-Sons, New York (1973)
3. Rao, C.R.: Linear Statistical Inference and Its Applications, 2nd edn. Wiley Interscience, Hoboken (2002)
4. Glover, F., Laguna, M.: Tabu Search, in Modern Heuristic Techniques for Combinatorial Problems. In: Reeves, C.R. (ed.). John Wiley & Sons, Inc., Chichester (1993)
5. Fisher, R.A.: The Statistical Utilization of Multiple Measurements. Annals of Eugenics 8, 376–386 (1938)
6. Baudat, G., Anouar, F.: Generalized Discriminant Analysis Using a Kernel Approach. In: Neural Computation, vol. 12, pp. 2385–2404 (2000)
7. Loog, M., Duin, R.P.W., Haeb-Umbach, T.: Multiclass Linear Dimension Reduction by Weighted Pairwise Fisher Criteria. IEEE Trans. on Pattern Analysis and Machine Intelligence 23(7), 762–766 (2001)
8. Bolshakova, N., Azuaje, F.: Cluster Validation Techniques for Genome Expression Data. In: IEEE Conference on Signal Processing, pp. 825–833 (2003)
9. Davies, D., Bouldin, D.: A Cluster Separation Measure. IEEE Trans. Pattern Anal. Mach. Intell. 1(2), 224–227 (1979)

A Novel Hybrid Structure for Clustering

Mahdi Yazdian Dehkordi[1], Reza Boostani[1], and Mohammad Tahmasebi[2]

[1] Computer Science and Engineering Department, Shiraz University, Shiraz, Iran
[2] Computer Engineering Department, Yazd University, Yazd, Iran
yazdian@cse.shirazu.ac.ir, boostani@shirazu.ac.ir,
tahmasebi@yazduni.ac.ir

Abstract. Conventional clustering methods such as k-means or subtractive clustering need some information include the optimal number of clusters, otherwise, they lead to the poor results. However in many practical situations it is often so difficult to determine the best number of clusters. On the other hand, there are other clustering algorithms such as Rival Penalized Competitive Learning (RPCL) or ISODATA which find the number of clusters automatically. But these clustering methods have a problem in locating the cluster centers appropriately. In this paper, a novel hybrid structure is proposed which uses these two types of clustering algorithms as complementary to improve clustering performance. Moreover, a new weighted version of RPCL algorithm which is called here WRPCL is suggested. This structure not only outperforms the choice of the optimal number of clusters but also improves the performance of the overall clustering result.

Keywords: Clustering, RPCL, Subtractive, k-means, Min-Max, Hybrid.

1 Introduction

Clustering algorithms can be divided into two main categories regarding the number of clusters. Some methods need to pre-define the number of clusters, while other methods determine it dynamically [1]. We call the first group as static clustering (SC) and other one dynamic clustering (DC). SC methods like k-means work well when the number of clusters determined properly. However, in many practical situations it is hard to know the exact number of clusters. To overcome this problem DC methods are proposed.

ISODATA [1], [5] determines the cluster number by merging and splitting clusters iteratively but its performance is very sensitive to its parameters, and also the number of parameters is so high. The original RPCL clustering algorithm [3], [4] rewards the winner cluster center similar to k-means. Moreover, it penalizes the other cluster centers as rivals. The only parameter in this algorithm is de-learning rate [3].

The key problem of DC methods is locating the cluster centers improperly. For example, since the RPCL algorithm tries to push extra cluster centers out of the data region, it can not find the location of cluster centers suitably. ISODATA clustering also suffers form this problem, because of weak splitting and merging criteria [5].

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 888–891, 2008.

In this paper, a new hybrid structure is proposed to improve the clustering result, using the clustering methods as a complementary form. Furthermore, we propose a weighted-RPCL algorithm to find the number of clusters.

The reminder of this paper is organized as follows. In section 2 the original RPCL is described. Section 3 represents the proposed weighted RPCL. Subtractive clustering is provided in section 4. In section 5, our novel hybrid structure is explained. The result of the suggested WRPCL and the novel structure are discussed in section 6. Finally the conclusion part is presented.

2 Original RPCL

RPCL algorithm can be regarded as an adaptive version of k-means. In each iteration of RPCL a pattern randomly is selected then the nearest cluster center rewarded as winner and the other centers are punished. The change of location for each cluster center would be calculated by the following formula.

$$\Delta W^p = \begin{cases} \alpha_c (X^n - W_t^p) & if\ p = c(n) \\ -\alpha_r (X^n - W_t^p) & if\ p \neq c(n) \end{cases} \tag{1}$$

Where, the parameter a_c and a_r are the learning and de-learning rates, respectively. Parameter a_r must be much less than a_c to achieve correct convergence. According to this, a_c and a_r would be set as equations (2) and (3).

$$\alpha_c = \alpha_0 / (c_1 t + c_0) \tag{2}$$

$$\alpha_r = \alpha_c \times f(X^n, W_t^p)^{-R-2} \tag{3}$$

Where α_0, c_0, c_1 are positive constants which in this paper set as 0.04, 1, 0.015. The parameter R and t show de-learning rate constant and iteration number.

3 Novel Weighted RPCL

RPCL method can be implemented with a number of rivals between 1 and K_0-1which K_0 is the number of clusters [4]. If all loser cluster centers are penalized, it is known as DSRPCL1 but if only the first loser is penalized, it is called DSRPCL2.

Finding the optimal number of rivals for each especial dataset is too difficult. For this reason we penalize the all losers but using different weights. We called this method as Weighted-RPCL, i.e. WRPCL. Let $\{W^1, W^2, \ldots, W^L\}$ as all cluster centers which must be punished. In this case, the equation (3) changed to (4).

$$\alpha_r = \alpha_c \times \mu^{n,p} \times f(X^n, W_t^p)^{-R-2} \tag{4}$$

The $\mu^{n,p}$ shows the weight of cluster center W^p from sample X^n which is as follows.

$$\mu^{n,p} = \{\underset{i=1}{\overset{L}{Max}}\, f(X^n, W^i) - f(X^n, W^p)\} \,/\, \{\underset{i=1}{\overset{L}{Max}}\, f(X^n, W^i) - \underset{i=1}{\overset{L}{Min}}\, f(X^n, W^i)\} \tag{5}$$

Where i and L are the index and number of rivals, receptively.

4 Subtractive Clustering

In Subtractive clustering algorithm, samples are considered as the candidates for cluster centers. At first a density measure calculated for each sample. Afterward, the sample with the largest value of density measure selected as first cluster center and the density of the samples reduced. This process repeated for next cluster center [2].

5 Novel Hybrid Structure

There are several clustering algorithms which find the number of cluster automatically but locate the cluster centers unsuitably. Besides, other clustering methods suppose a pre-defined clusters number but locate cluster centers better than the first group. Therefore, two clustering algorithms can be used as a complementary form as follows.

Step 1: Normalize the data to remove improper effect of different scales of features.
Step 2: Use a SC method to initialize the cluster centers of a DC method.
Step 3: Find the best cluster number (N^F) by a DC method which is initialed with resulted cluster centers in step 2.
Step 4: Apply a SC method with N^F cluster number due to the final cluster centers. If the SC algorithm in this step is sensitive to the initial point it would be initialized with the resulted cluster centers from step 2 or 3.

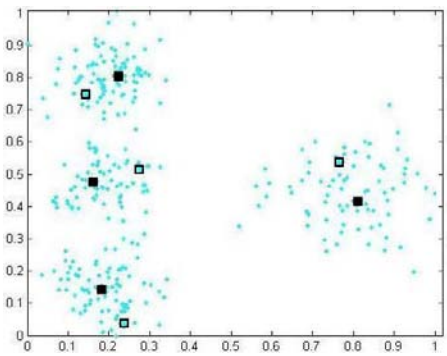


Fig. 1. Artificial dataset. Squares show 8 cluster centers found by subtractive clustering. Solid squares are the 4 first centers found.

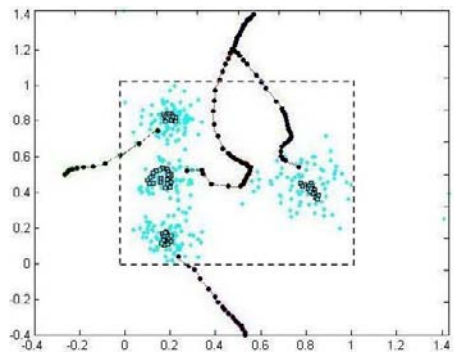


Fig. 2. WRPCL throws out extra cluster centers. Dashed lines show the boundary of the data.

6 Experimental Results

It is useful to summarize the results of novel hybrid structure. The step 3 involves a DC algorithm to find the number of clusters. ISODATA has many parameters and is so sensitive to them; while WRPCL has only one parameter, i.e. de-learning rate. Hence, WRPCL is utilized to find the number of clusters. Fig.2 visualizes how

WRPCL throws out extra cluster centers. Before that, in step 2, a SC method must be selected to initialize WRPCL. If the initial points distribute in dense places, it would be suitable. Therefore, subtractive clustering is utilized in step 2. Table 1 and 2 summarize the result of the proposed new hybrid structure. The classification rate is used to measure the performance of each clustering method [1], [5]. For final clustering in step 4, we employed the number of clusters which is resulted from step 3 (N^F) and passed it to k-means, subtractive and Min-Max clustering.

Table 1. Result with Euclidean distance

Dataset / Algorithm	Iris (#Cluster=5)	Wine (#Cluster=5)	Glass (#Cluster=9)
WRPCL	88.28±9.12	94.64±2.04	40.54±5.03
k-means	88.89±7.11	95.57±3.44	59.07±10.08
Subtractive	**96**	**96.07**	**66.36**
Min-Max	82.04±13.96	84.46±20.42	50.12±8.76

Table 2. Result with Manhattan distance

Dataset / Algorithm	Iris (#Cluster=5)	Wine (#Cluster=5)	Glass (#Cluster=9)
WRPCL	87.90±1.20	91.22±6.46	57.33±8.62
k-means	89.85±23.19	**94.85**±4.9	63.41±12.01
Subtractive	**96**	92.13	**66.82**
Min-Max	85.37±10.04	86.47±13.44	51.08±6.69

Achieved results illustrate that subtractive clustering overcomes the other clustering methods. Moreover, it is stable while other methods (especially k-means) have much tolerance. Altogether we propose following clustering methods for novel hybrid structure. At first, subtractive clustering would be used to initialize WRPCL (Fig.1 squares). Afterward, the number of clusters from WRPCL and first N^F cluster centers from subtractive clustering will be used to find the final clustering results (Fig.1 solid squares). Manhattan distance is preferred to use as distance measurement.

7 Conclusion

In this paper, a novel hybrid structure has been proposed which used static and dynamic clustering methods as complementary form to improve the clustering result. For dynamic clustering methods, a new weighted version of RPCL algorithm called WRPCL has been proposed to find the number of clusters in input datasets.

The proposed structure not only outperforms the selection of the optimal number of clusters but also improves the performance of the overall clustering result.

References

1. Jain, A.K., Dubes, R.: Algorithms for Clustering Data. Prentice-Hall, Englewood Cliffs (1988)
2. Jang, J.S., Sun, C.T., Mizutani, E.: Neuro-Fuzzy and Soft Computing. Prentice-Hall, Englewood Cliffs (1997)
3. Cheung, Y.M.: Rival Penalization Controlled Competitive Learning for Data Clustering with Unknown Cluster Number. In: 9th ICONIP, Singapore, vol. 2, pp. 467–471 (2002)
4. Ma, J., Wang, T.: A Cost-Function Approach to Rival Penalized Competitive Learning. IEEE Transaction on Systems, Man and Cybernetics 36, 722–737 (2006)
5. Taheri, M., Boostani, R.: Novel Auxiliary Techniques in Clustering. In: Proc. of Conf. WCE 2007, London. Lecture Notes in Eng. & CS., vol. 1, pp. 243–248 (2007)

6R Robots; How to Guide and Test Them by Vision?

Azamossadat Nourbakhsh[1] and Moharram Habibnezhad Korayem[2]

[1] Islamic Azad University, Lahijan, Iran
A.S.Nourbakhsh@gmail.com
[2] Robotic Research laboratory, Mechanical Engineering Department, Iran University of Science and Technology, Tehran, Iran
hkorayem@IUST.AC.IR

Abstract. The aim of visual servoing is to control robots by using the data obtained through vision. In this article, Feature-based and Position-based methods of visual servoing are used in visual servoing simulator of a 6R robot. In this simulator, three cameras were used simultaneously. The camera which is installed as eye-in-hand on the end-effector of the robot is used for visual servoing in a Feature-based method. The target object is recognized according to its characteristics and the robot is directed toward the object in compliance with an algorithm similar to the function of human's eyes. Then, the function and accuracy of the operation of the robot are examined through Position-based visual servoing method using two cameras installed as eye-to-hand in the environment. Finally, the obtained results are tested under ANSI-RIA R15.05-2 standard.

Keywords: 6R Robot, camera, visual servoing, Feature-based visual servoing, Position-based visual servoing, Performance tests.

1 Introduction

In this project, the operation of recognizing the object by camera is programmed exactly similar to recognition done by human's eyes. Visionary systems are often classified in terms of the number of cameras and their position. Using camera as a stereo structure (such as observing the environment using two cameras) eases many of visual problems of computers. Gregory Flandin has explained the manner of using cameras in eye-in-hand/eye-to-hand ways in controlling robots [3]. In eye-in-hand method, the camera is set on the end-effector, while in eye-to-hand method the camera observes the entire environment of the robot and is installed outside the robot. Also controlling through vision and test of a 3P labeler robot is designed and simulated by Korayem et al. [1]. In this method, a camera which is set on the end-effector of the robot is used for finding the target and the end-effector is controlled to reach the target using a Feature-based visual servoing method. Two cameras which are placed in the environment and a Position-based visual servoing method have been also used in order to test the robot. There is also software proposed for simulating the function of a 6R robot in Position-based visual servoing [2]. In this software, two cameras are used

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 892–896, 2008.

which are placed in the environment in a specific distance. The pictures taken by the cameras are seen in the picture plan. Then, considering the recognition of the end-effector and the object, it is tried to reduce their location difference to zero.

In this simulator, which is the second version of the simulating software of 6R robot, measures are taken in order to create a Feature-based vision for controlling the robot. Three cameras are used simultaneously. The operation of controlling the robot is done through Feature-based visual servoing, and with respect to the characteristics of the target object and their recognition by the robot, accessibility to the targets becomes plausible. The considered characteristics in this project are color and the shape of objects. Also, Position-based method is used in order to test the performance of the robot. In this case, two cameras which are installed in the environment are used. After analysis of the picture and recognition of the target object and the end-effector, their locations in the coordinate of the picture are estimated, and then the visionary system moves the end-effector toward the target object. Also the movement of the robot in different Paths is tested and the obtained results are tested under ANSI-RIA R15.05-2 standard.

2 Introducing the 6R Robot and the Simulator

The robot which is designed in the University of Science and Industry has 6 degrees of freedom and all of its joints are of rotational kind. As seen in Fig. 1. , it has 3 degrees of freedom in its body, shoulder and head and also 3 degrees of freedom in the end-effector which perform the roll, pitch and yaw rotations. The first arm rotates horizontally around the vertical axis. The second arm rotates in an axis which is vertical comparing to the axis around which the first arm rotates, and the third junction rotates in a parallel line comparing to the second junction.

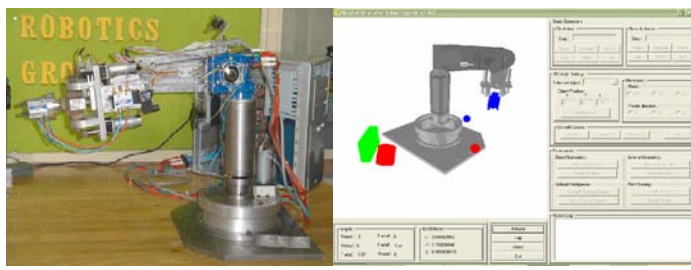


Fig. 1. structure of a 6R robot and the simulator software used for visual servoing

One of the aims of this project is to employ a set of issues related to the vision on machines, from the way of taking pictures by camera to processing the picture and recognizing the predetermined object and finally controlling the robot by vision, and this is why a 3D environment for the project of simulation of a 6R robot has been created. In this software, 3 cameras have been used. The user can do the monitoring job by switching to any of these cameras at any moment. By the camera which is installed on the end-effector, the simulation software receives the data exactly similar to the real situation and after processing the pictorial data, seeks an object with the

characteristics of the target object (shape and color). The aim of installing two cameras in the environment is observance of the entire space in which the robot is placed by the observer. These cameras are not fixed and can be moved by the user in any point of the space. The camera number 1 looks at the robot from the right side and the camera number 3 observes the robot from its front side. In this software, it is possible for the object to move in accordance with the different planes of coordinate axes. In this condition, the location of the selected object in the picture plan can be changed using the left button of the mouse. Also each of the selected objects rotates around X, Y or Z axes. This function is performed by the right button of mouse.

3 Visionary Operation of the Robot

In controlling the robot via vision, we consider two principal operations: first is the ability of the robot in performing the desired operation, and then bringing the test operation into action. In this software, using a Feature-based method, the robot is controlled to reach the target object. In this condition, there is one camera which is installed on the end-effector of the robot, and the pictures taken by this camera are analyzed. The distinguishing process is done based on two characteristics of color and shape. As seen in Fig.1., There are different objects in the working space of a robot. First, one picture is received from the camera installed on the end-effector of the robot. After processing the picture and recognition of the object by the visionary system, the end-effector of the robot is directed toward the object. In this step, the applied algorithm works exactly similar to human's eyes and checks the border points of the picture. After some steps, the center of the object will be coincided with the center of the picture. In this condition, it's time to adjust the distance between the end-effector and the object. In this condition, the visionary system moves in maximum 15 steps to catch the object. Pictures of the different stages of this process are taken by cameras number 1 and 2.

In the next stage, using a Position-based method, we have put our efforts to execute the movement of the robot for catching the object. First, locations of the target object and the end-effector are distinguished by cameras number 1 and 3 which are placed in the environment. Since there is a need for the mapping of the picture plan coordinate for the reference coordinate in order to have the locations of the end-effector and the object in the Global Reference Coordinate, a neural network system is used instead of mapping. By this neural network system, the two dimensional data of location of objects acquired by the pictures of the two cameras, are converted to three dimensional data of location which is recognizable by the robot. Then, by solving Inverse Kinematics equations, the robot tries to reduce the location difference of the target object and the end-effector to zero.

4 Performance Tests

In this method, the end-effector of the robot is moved on a certain Path. Two cameras observe the end-effector from outside. In this case, three different Paths of line, circle and rectangle are tested in the movement of the robot.

The important issue is that the proposed formulas for the robot (based on the Denavit-Hartenberg symbolization) are calculated for a situation where Z axis faces upward (in other words, where the machine is left-handed), but Direct X uses a right-handed axis and so Y axis faces upward. Hence, two quantities of X and Z are considered. Here, the diagram of movement of the robot in three Paths is observed:

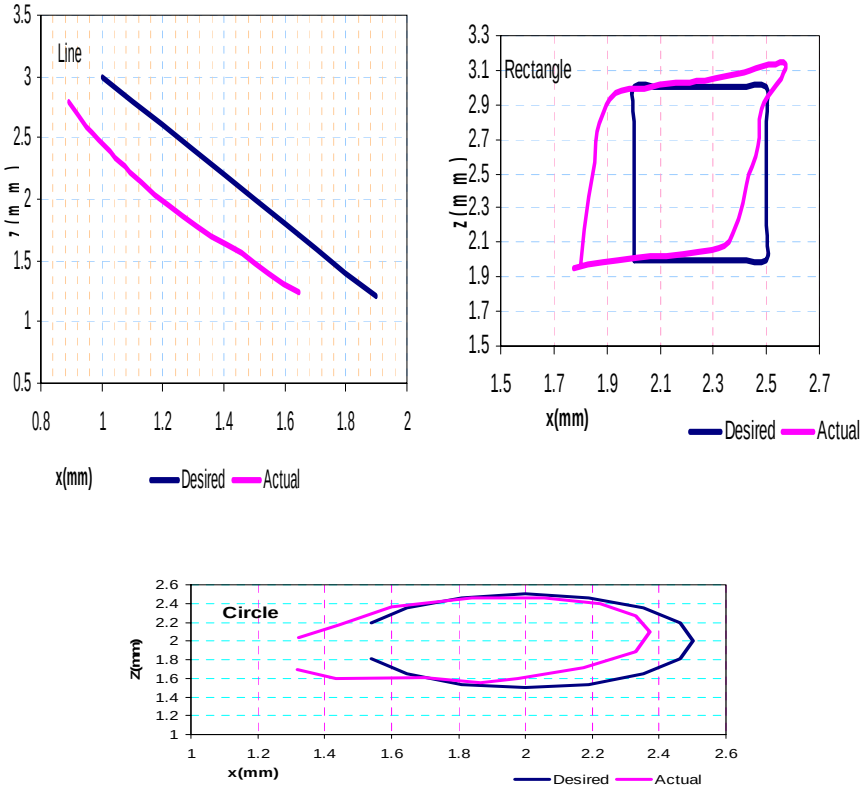


Fig. 2. Diagrammed comparison between optional quantities and quantities attained by the system in Linear, Rectangular and Circular Paths

5 Conclusion

In this piece of text, use of a visionary system in controlling a 6R robot was simulated. A combined version of Feature-based and Position-based methods of visual servoing is produced and introduced in simulator software and this software has been able to perform visual servoing of a robot with a good deal of proximity to reality, using a 3D graphic environment. Possibility of using 3D facilities in movement and rotation of any object in the environment, contributes a lot to comprehension of 3D environment and becoming familiar with it. Also the pictures received from the

cameras are saved in order to be processed by user. Then, Feature-based method was applied in order to control the robot. In this stage, the characteristics of color and shape of objects were used in the environment. Since the camera installed on the end-effector was used in this method, a controlling algorithm similar to the function of human's eyes was employed.

In order to perform performance tests in accordance with the existing standards, there is a need for applying two cameras placed in the environment for processing the data and obtaining the location of the end-effector relative to the reference coordinate (ground) and in fact the two cameras did it together. In this project, from among different existing standards, ANSI-RIA standard is observed and under the fixed norms of ANSI-RIA standard statistical and scientific analysis of the robot is made.

References

1. Aliakbarpour, H.: Vision Simulating and introducing an algorithm for Image Processing and using in the Robot, Thesis for taking M. S., Science and Researches Branch of Islamic Azad University (1999)
2. Heidari, F.S.: Simulation of Visual Servo Control and Performance Tests of the 6R Robot Using Position Based and Image Based Approach, Thesis for taking M. S., Iran University of Science and Technology (October 2006)
3. Flandin, G.: Eye-in-hand/Eye-to-hand cooperation for Visual Servoing. IEEE Press, Los Alamitos (2000)
4. American National Standard for Industrial Robots and Robot Systems Path-Related and Dynamic Performance Characteristics Evaluation. ANSI/RIA R15.05-2 April 16 (2002)

Hierarchical Diagnosis of Vocal Fold Disorders

Mansour Nikkhah-Bahrami[1], Hossein Ahmadi-Noubari[2], Babak Seyed Aghazadeh[1], and Hossein Khadivi Heris[1]

[1] Mechanical Engineering Department, University of Tehran, Tehran, Iran
[2] Electrical and Computer Engineering Department, University of Tehran, Tehran, Iran
b.aghazadeh@me.ut.ac.ir

Abstract. This paper explores the use of hierarchical structure for diagnosis of vocal fold disorders. The hierarchical structure is initially used to train different second-level classifiers. At the first level normal and pathological signals have been distinguished. Next, pathological signals have been classified into neurogenic and organic vocal fold disorders. At the final level, vocal fold nodules have been distinguished from polyps in organic disorders category. For feature selection at each level of hierarchy, the reconstructed signal at each wavelet packet decomposition sub-band in 5 levels of decomposition with mother wavelet of (db10) is used to extract the nonlinear features of self-similarity and approximate entropy. Also, wavelet packet coefficients are used to measure energy and Shannon entropy features at different spectral sub-bands. Davies-Bouldin criterion has been employed to find the most discriminant features. Finally, support vector machines have been adopted as classifiers at each level of hierarchy resulting in the diagnosis accuracy of 92%.

Keywords: Hierarchical classification, wavelet packets, nonlinear signal analysis, vocal fold disorders.

1 Introduction

Physiological alterations of vocal cords cause unhealthy patterns of cords' vibration and the decrease in patients' speech signal quality known as voice pathologies. Therefore, the detection of incipient damages to the cords is useful in improving the prognosis, treatment and care of such pathologies. Most of the recent computer-based algorithms for automatic screening are based on wavelets, fractals, neural maps and networks. Guido *et al* [1] have tried different wavelets on the search for voice disorders. Mother wavelet of Daubechies with support length of 20 (db10) was found as the best wavelet for speech signal analysis among commonly used mother wavelets. Local discriminant bases (LDB) and wavelet packet decomposition have been used to demonstrate the significance of identifying the signal subspaces that contribute to the discriminatory characteristics of normal and pathological speech signals in a work by Umapathy *et al* [2]. Matassini *et al* [3] have analyzed voice signals in a feature space consisting quantities from chaos theory (like correlation dimension and first Lyapunov exponent) besides conventional linear parameters among which nonlinear parameters have reported to have clear separation between normal and sick voices.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 897–900, 2008.

In this work a novel approach has been used to diagnose different common types of vocal fold disorders. After discrimination of normal and pathological signals at the top level of hierarchy, pathological voices have been categorized regarding their origin of pathology as neurogenic and organic disorders. In the following organic disorders of vocal fold nodules and polyp have been identified at the lower level.

The rest of this paper is organized as follows: In section 2 and 3 materials and datbse used in this study are presented. Experimental results are discussed in section 4. Finally, section 5 presents the conclusions.

2 Methods

The Davies-Bouldin (DB) criterion has been proven to be effective in many biomedical applications when used to evaluate the classification ability of feature space [4]. The DB index (DBI), or cluster separation index (CSI) is based on the scatter matrices of the data and is usually used to estimate class separability. In this paper, DBI has been employed to assess two-class separability of features at each level of hierarchy.

Approximate entropy (ApEn) is a statistical index that quantifies regularities and complexity which has application to a wide variety of relatively short and noisy time series data [5]. The following characteristics make ApEn a useful tool in biomedical signal analysis: 1- A robust estimate of ApEn can be obtained by using short data; 2-It is highly resistant to short strong transient interference; 3- The influence of noises can be suppressed by properly choosing the parameters in the algorithm; 4- It can be applied to both deterministic and stochastic signals and to their combination.

Although different definitions of second-order self-similarity can be found in the literature, they share the common idea of processes which do not change their qualitative statistical behavior after aggregation. The Hurst exponent (H) characterizes the level of self similarity, providing information on the recurrence rate of similar patterns in time at different scales. Several methods are available to estimate the Hurst parameter. In this paper, wavelet based scaling exponent estimation has been employed to calculate Hurst exponent [6].

In-depth mathematical descriptions of the methods used in this study are available in references.

3 Database

Used in this study are sustained vowel phonation samples from subjects from the Kay Elemetrics Disordered Voice Database [7]. It includes signals from 53 normal voices, 54 unilateral vocal fold paralysis, 20 vocal fold polyp, and 20 vocal fold nodules. This represents a wide variety of organic and neurogenic voice disorders. Subjects were asked to sustain the vowel /a/ and voice recordings were made in a sound proof booth on a DAT recorder at a sampling frequency of 44.1 kHz.

4 Results and Discussion

For each voice sample, energy and Shannon entropy of WP coefficients in addition to self-similarity and approximate entropy of the reconstructed signals at each sub-band have been calculated to construct the feature vector of length 252. Next, DBI of each potential feature out of 252 has been measured for corresponding data samples at each level in order to quantify its 2-class separability. For each level of hierarchy, having the feature vector sorted in terms of DB indices, one can easily construct the effective feature vector of length *n* by using first *n* features which possess least DB values.

To achieve the highest accuracy of classification 8 most discriminant features have been selected to construct the feature vector. Table 1 demonstrates participating features at each level of hierarchy. According to the Table, energy of signals at different wavelet packet decomposition sub-bands is the most efficient feature in discrimination of normal/pathological voices. The average DB index of features at level two is highest among the levels of hierarchy which shows the difficulties in the classification of neurogenic and organic vocal disorders. The overall number of contributing feature types at each level of hierarchy is depicted in fig 1.

Table 1. Participating features at levels of hierarchy

Level of hierarchy	Selected Features and their corresponding nodes	Average DBI
1	Energy at:32,34,16,0,15, Self-similarity at: 0 ApEn at 34,8	1.25
2	Energy at:10,45,21,43,22,7 Shannon entropy at: 45, ApEn at 34	3.05
3	Self-similarity at: 32,15,18,45,26, ApEn at: 18,56,37	1.72

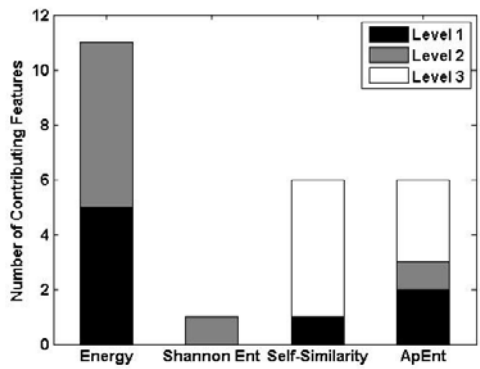


Fig. 1. Overall contribution of features

Although many researchers have used linear features such as energy and Shannon entropy in normal/pathological discrimination problems, nonlinear features with significant influence in discrimination of mass related vocal fold disorders (i.e. nodules and polyp) seem unavoidable in a more comprehensive vocal fold disorders study.

Moreover, selected sub-bands are distributed over the whole available frequencies, which shows that pathological factors do not influence specific frequency range and accentuates the role of WP decomposition. The privilege of hierarchical diagnosis approach is evident regarding different selected features at each level of hierarchy.

In the following, support vector machines have been used for classification aim. Randomly chosen 20 percent of voice signals have been used to test the classifiers. To improve the classification generalization the procedure above has been repeated and the mean accuracy has been considered.

The classification accuracy of 92 percent among test samples shows that hierarchical approach for diagnosis of vocal fold disorders is satisfyingly reliable.

5 Conclusion

In contrary to the situation in which the main target is to discriminate normal and pathological cases, the feature variations in the cases of vocal fold disorders due to their similar effects on the vibration pattern of vocal folds, make some difficulties in classification. In this paper, a hierarchical approach has been proposed for classification of voice signals into 4 classes of normal, unilateral vocal fold paralysis, vocal fold polyp, and vocal fold nodules. Regarding nonlinear nature of voice production and chaotic characteristics of voice pathologies, the high percentage of participating nonlinear features in discrimination of different voice pathologies is expectable. Moreover, according to difference in selected best features at levels of hierarchy, it can be concluded that a hierarchical diagnosis procedure is effective and efficient.

References

[1] Guido, R.C., Pereira, J.C., Fonseca, E., Sanchez, F.L., Vierira, L.S.: Trying different wavelets on the search for voice disorders sorting. In: Proceedings of the 37th IEEE International Southeastern Symposium on System Theory, pp. 495–499 (2005)

[2] Umapathy, K., Krishnan, S.: Feature analysis of pathological speech signals using local discriminant bases technique. Med. Bio. Eng. Comput. 43, 457–464 (2005)

[3] Matassini, L., Hegger, R., Kantz, H., Manfredi, C.: Analysis of vocal disorders in a feature space. Medical Engineering & Physics 22, 413–418 (2000)

[4] Wang, G., Wang, Z., Chen, W., Zhuang, J.: Classification of surface EMG signals using optimal wavelet packet method based on Davies-Bouldin criterion. Med. and Biol. Eng. Comput. 44, 865–872 (2006)

[5] Akay, M.: Nonlinear Biomedical signal processing, 2nd edn., vol. 2, p. 72. IEEE Press, Piscataway (2000)

[6] Veitch, D.N., Taqqu, M.S., Abry, P.: Meaningful MRA initialization for discrete time series. Signal Processing 80(11), 1971–1983 (2000)

[7] Disordered voice database (CD-Rom), Version 1.03, Massachusetts Eye and Ear Infirmary, Kay Elemetrics Corporation, Boston, MA, Voice and Speech Lab (October 1994)

A Voronoi-Based Reactive Approach for Mobile Robot Navigation

Shahin Mohammadi[1] and Nima Hazar[2]

[1] Computer Science Department, University of Tehran, Tehran, Iran
mohammadi@sadi.ut.ac.ir
[2] Computer Engineering Department, Sharif University of Technology, Tehran, Iran
hazar@ce.sharif.edu

Abstract. Robot navigation is a challenging problem in robotics, which involves determining of robot positioning relative to objects in the environment, and also the mobility of robot through obstacles without colliding into them. The former is known as localization, while the latter is called motion planning. This paper introduces a roadmap method for solving motion planning problem in a dynamic environment based on Generalized Voronoi Diagram (GVD). The efficiency of the proceeding work is verified by examining it in a sample home environment.

1 Introduction

Autonomous robots are those which can perform desired tasks in unstructured environments without continuous human guidance. Building of such autonomous robots has been a primary goal in the field of robotics and artificial intelligence over the past decade. One of the main challenges of this field, is the navigation problem. In this paper the Generalized Voronoi Diagram (GVD) is used to build a topological map of the environment, in order to minimize the risk of colliding with obstacles during navigation.

The remainder of this paper is organized as follows. Section 2 presents the solution which is used for building the roadmap and mapping objects to its vertices. Section 3 will describe the motion planning algorithm that is implemented on our robot. The final section will cover the conclusion and the experimental results, which are based on our experiments in Robocup @Home competitions.

2 Building Roadmap

The building block of GVD is the points equidistance to at least two obstacles [1]. In order to compute these points, it is needed to develop a function to find the closest obstacle to each point in the field. The following pseudo code takes initial position *s* and uses the region growing method to find closest obstacle, in addition to its distance and angel from the source point:

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 901–904, 2008.

Algorithm 1. Find Closest Obstacle

```
1: Q.Enqueue(s)
2: while !Q.Empty() do
3:     v = Q.Dequeue()
4:     for u ∈ Neighbour(v) do
5:         if u.visited == false then
6:             if u.status == FREE then
7:                 u.visited = true
8:                 Q.Enqueue(u)
9:             else
10:                Angle = Arctg(u.y−s.y / u.x−s.x)
11:                Distance = √((u.x−s.x)^2 + (u.y−s.y)^2)
12:                Return(Angle, Distance, u.ID)
13:            end if
14:        end if
15:    end for
16: end while
```

Algorithm 2 uses the Find Closest Obstacle(FCO) function and compute the GVD based on a probabilistic approach (See [2] and [3]). In every iteration of the algorithm, a random point u is selected from the uniform distribution and then it is retracted to the Voronoi diagram. First the closest obstacle point to u is found using FCO. Afterwards, an iterative method is used to scan the points in the opposite direction until the closest obstacle to t_k differs from the closest obstacle to t_{k-1}. Now the point which has same distance from its two nearest obstacles can be found by using the binary search between t_k and t_{k-1} with a desired precision ε [4]. After adding this point to the Voronoi vertices list, the algorithm continues executing until it has N vertices. The following pseudo-code uses the FCO method and generates N points on the GVD:

Algorithm 2. GVD Builder

```
1: V ← E ← ∅
2: for k = 1 to N do
3:     u ∈ U[0,w] * U[0,h] and u.status = FREE
4:     Δx = cos(180 + Angle) * Step
5:     Δy = sin(180 + Angle) * Step
6:     ObjectID2 = ObjectID1
7:     while ObjectID1 == ObjectID2 do
8:         s ← t
9:         (t.x, t.y) += (Δx, Δy)
10:        (Angle, Step, ObjectID2) ⇐ FCO(t)
11:    end while
12:    Vertices += Binary_Search(s, t, ε);
13: end for
14: UpdateEdges(Vertices)
```

Each node in the generated map is marked with an object ID. This is not sufficient, since getting very close to an object is not safe. Therefore it is needed to define the "object neighborhood" concept. For each object, its neighborhood is defined as:

$$N_{Obj} = \{u \in V_{GVG} | \exists v \in boundry_{Obj}; Dist(u,v) < \delta\} \quad (1)$$

Object neighborhood is used to query vertices set near to an arbitrary object.

3 Motion Planning

3.1 Updating Roadmap

There are three parameters describing the state of each edge during the execution:

- **Distance** Distance of each edge e, D_e, is the euclidian distance between its two incident vertices. It is used to find out the shortest path.
- **Risk Factor** Risk factor, as the name describes itself, is the parameter that measures the reliability of a path. Risk factor of each node v, R_v, is defined as the inverse of minimum distance from that node to obstacles in the field. Risk factor of each edge e is initialized as:

  $$R_e = \max(R_{v1}, R_{v2}) * Robot_{radius} \quad (2)$$

  where $v1, v2$ are the incident vertices to the edge e.
  During each iteration, if there were no obstacle in front of the robot, it keeps the old risk factor. Otherwise the risk factor can be calculated using the following formula:

  $$R_e = \max(\frac{1}{\delta_{Left}}, \frac{1}{\delta_{Right}}) * Robot_{radius} \quad (3)$$

  Where the risk factor lower than one, means a blocked path.
- **Blocking Factor** Blocking factor is the parameter which indicates the probability of visiting an obstacle on an edge. Blocking factor for edge e is calculated the last time the robot sensed an obstacle on that edge which blocked the path, t_e, using function:

  $$B_e = \frac{e^{\alpha - \frac{t_e}{\tau}}}{1 + e^{\alpha - \frac{t_e}{\tau}}} \quad (4)$$

  α is the blocking time and τ is the repair rate. The α will adjust the minimum time which the edge is unusable; while the τ will adjust its repair rate.

3.2 Path Planning

Given an initial position, the nearest Voronoi vertex to that position is located and set as the temporary destination. After reaching each the temporary destination, the robot updates the parameters of the all visible edges, namely their risk factor R_e and blocking factor B_e. Next, it calculates the overall weight of those edges using the following formula:

$$Weight = \alpha * D_e + \beta * R_e + \gamma * B_e \quad (5)$$

α, β and γ must be chosen based on the environment properties. For an indoor robot safety is the most important factor in mind. In order to satisfy this properties, α, β and γ are chosen so that $\alpha < \beta < \gamma$;

Finally, the A* algorithm is used in order to recompute the shortest-safest path from current vertex, temporary destination, to the closest vertex which is mapped to the desired object, final destination.

4 Experimental Results

To illustrate the effectiveness of the described algorithm, three tests with three different scenarios are performed. In all of them the robot is placed in an initial position on the map and had to safely navigate around objects without bumping into obstacles. The destination could be any place or object in the map with a minimum distance of $4m$, and reaching an object means obtaining a maximum distance of $50cm$ from the object. A complete list of objects which could used to navigate is provided before the competition.

In the first test, the initial position of the robot was predetermined. The robot was placed on a Voronoi vertex and commanded to navigate to the fridge. In the second test, both the start and end position were unknown. The robot was placed beside the fridge, and is commanded to navigate to the table. Finally in the third test, the start position of the robot was the left and down corner of the map and the destination was the bookshelf. There were two dynamic obstacles in the map. The first one was a cube which blocked the path between the table and the TV, and the other one was a small toy which placed near to lunch table but did not blocked the path completely.

the robot successfully passed Phase I& II. In the Final Phase, the robot decided to round the lunch table, in order to avoid colliding into the toy and this is due to increasing in the R_e's value, which in turn affected the edge weights. It is notable here that the efficiency of the algorithm depends vastly on the choice of α, β and γ.

References

1. Choset, H., Konukseven, I., Burdick, J.: Mobile robot navigation: Issues in implementation the generalized voronoi graph in the plane. In: Proceedings of the 1996 IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems, pp. 241–248 (1996)
2. Kavraki, L.E., Latombe, J.C.: Randomized preprocessing of configuration space for fast path planning. In: ICRA, pp. 2138–2145 (1994)
3. Overmars, M.H., Švestka, P.: A probabilistic learning approach to motion planning. In: WAFR: Proceedings of the workshop on Algorithmic foundations of robotics, Natick, MA, USA, pp. 19–37. A. K. Peters, Ltd. (1995)
4. Nieuwenhuisen, D., Kamphuis, A., Overmars, M.M.M.H.: Automatic construction of high quality roadmaps for path planning. Technical Report UU-CS-2004-068, Institute of Information and Computing Sciences, Utrecht University (2004)

Evaluation of PersianCAT Agent's Accepting Policy in Continuous Double Auction, Participant in CAT 2007 Competition

Sina Honari, Amin Fos-hati, Mojtaba Ebadi, and Maziar Gomrokchi

[1] Premier Ideas Support Center, School of Engineering, Shiraz University, Shiraz, Iran
{Sina2222,amin.fos.hati,mojtaba3000,mgomrokchi}@gmail.com

Abstract. As there is a growing tendency towards online auctions, online Multi Agent Systems for economic activities like stock exchanges is more in demand. CAT (CATallactics) competition has produced a great opportunity for researchers since 2007 to put their theories into practice in a real-time economic-based competition, combining traders and brokers. As one of participants, we evaluated our new accepting policy by putting it to challenge. In this paper we give a general overview of one of our policies in the market.

1 Introduction

As the demand increases for online auctions, the practice of using multi-agent systems within this framework is getting more attention. There has been a considerable research in the field of trading agents, but using this intelligent agent within an autonomous market with dynamic rules and policies is a new story. In other words, in addition to using autonomous trading agent, autonomous markets that facilitate trade for these traders, can produce more desirable markets. Running these autonomous agents in an online competition combining several markets and traders was a new idea that shaped a new game since 2007 called CAT [2] in Trading Agent Competition [3]. Market Design Scenario (CAT) has provided a new challenge for researcher to develop a market for traders as a framework for economic transactions. Since PersianCAT agent was one of the participants in this competition, here we try to analyze one of the policies of this agent called accepting policy by showing the performance of this policy in separate test runs.

2 Game Overview

The game consists of trading agents and Specialists (Brokers). Each specialist operates and sets the rules for a single exchange market, and traders buy and sell goods in one of the available markets in a double-sided auction [1].

Each trading agent has a trading strategy and a market selection strategy. The trading strategy is used to generate bids and asks (also called shouts) such as GD [4] and ZI-C [5] strategies, whereas the market selection strategy is used to select a specialist

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 905–908, 2008.

(market), examples are e-greedy [6] and softmax [6]. In the CAT competition ([7], [8]) the trading agents are provided by the CAT game, whereas specialists (and the rules of the markets) are designed by the entrants.

Each entrant is limited to operate a single market, which is achieved by implementing the following policies: Charging, Accepting, Pricing and Clearing Policies. Each specialist is assessed on three criteria [9] on each Assessment Day, and these criteria are then combined into a single score for that day. These criteria are: 1-profit: the profit obtained by a specialist as a proportion of the total profits obtained by all specialists. 2-Market Share: the number of traders registered with a specialist divided to all traders in the competition. 3-Transaction Success Rate (TSR): the proportion of bids and asks placed with a specialist which that specialist is able to match.

3 Accepting Policy

This policy determines which shouts are placed in the market. A specialist has the option to reject shouts which do not conform to the specialist's policy. Since in CAT game there are trading agents instead of human beings with different bidding strategies, the generated shouts may not produce transactions due to their unsuitable prices, so it may take a long time for incoming shouts to reach the equilibrium and make a transaction. What's more, since TSR is a scoring factor, this type of accepting can lead to lost of unmatched shouts. To solve this problem we introduce a new accepting policy called "Probabilistic Equilibrium Beating Accepting Policy".

In this policy the first shout in the auction is accepted by a probabilistic function. This function checks the possibility of transaction for new arriving shouts. If the possibility of the transaction is more than fifty percent shout is accepted. After accepting the first shout by this function we use Quote beating accepting policy. In this case we start each auction by a shout with higher chance of transaction but still give the market the chance to choose the final transaction price in a point between the first accepted shouts, so the market has enough flexibility for determining the final price and we reduce undesired shouts which don't lead to a transaction. In this policy we introduce three different sets of probabilistic equations:

$$\mathrm{P(a)} = \frac{\sum_{d \geq a} AT(d) + \sum_{d \geq a} B(d)}{\sum_{d \geq a} AT(d) + \sum_{d \geq a} B(d) + \sum_{d \leq a} BT(d) + \sum_{d \leq a} AN(d)} \tag{1}$$

$$\mathrm{P(b)} = \frac{\sum_{d \leq b} A(d) + \sum_{d \leq b} BT(d)}{\sum_{d \geq b} AT(d) + \sum_{d \leq b} A(d) + \sum_{d \leq b} BT(d) + \sum_{d \geq b} BN(d)} \tag{2}$$

$$\hat{P}(a) = \frac{\sum_{d \geq a} B(d)}{\sum_{d \geq a} B(d) + \sum_{d \leq a} BT(d)} \tag{3}$$

$$\hat{P}(b) = \frac{\sum_{d \leq b} A(d)}{\sum_{d \leq b} A(d) + \sum_{d \geq b} AT(d)} \tag{4}$$

$$\tilde{P}(a) = \frac{\sum_{d \geq a} AT(d) + \sum_{d \geq a} B(d)}{\sum_{d \geq a} AT(d) + \sum_{d \geq a} B(d) + \sum_{d \leq a} AN(d)} \quad (5)$$

$$\tilde{P}(b) = \frac{\sum_{d \leq b} BT(d) + \sum_{d \leq b} A(d)}{\sum_{d \leq b} A(d) + \sum_{d \leq b} BT(d) + \sum_{d \geq b} BN(d)} \quad (6)$$

$\sum_{d \geq a} B(d)$: The number of bids with prices higher or equal to 'a'

$\sum_{d \geq a} AT(d)$: The number of transacted asks with prices higher or equal to 'a'

$\sum_{d \leq a} BT(d)$: The number of transacted bids with prices lower or equal 'a'

$\sum_{d \leq a} AN(d)$: The number of accepted but not transacted asks with prices $\leq$ 'a'

$\sum_{d \leq b} A(d)$: The number of asks with prices less or equal to 'b'

$\sum_{d \leq b} BT(d)$: The number of transacted bids with prices less or equal to 'b'

$\sum_{d \geq b} BN(d)$: The number of accepted but not transacted bids with prices $\geq$ 'b'

$\sum_{d \geq b} AT(d)$: The number of transacted asks with higher or equal price to 'b'

For each pair of equations P(a) estimates the possibility of transaction for a new coming ask and P(b) checks this possibility for a new arriving bid. The difference between these three types of equations is the usage of different factors. In the first pair all the possible options have been used, while in the second pair only the factors in opposite type of shout have been used which means if this is a bid the possibility is checked by existing ask values and vise versa. In the third pair the factor are the same as GD [5] trader's bidding strategy.

We have compared these three types with the conventional Quote beating Accepting Policy in different test runs. For all agents we have used PersianCAT specialist with different accepting policies. The experiment was run for 200 trading days with 100 traders. We've conducted 3 separate runs. In the first run as you can see in Table 1 only GD[5] traders have been used. For second and third runs ZI-C [6] and Mixed (Combination of GD and ZI-C) traders have been used respectively (Table 2 and 3). In all tables PC1 represents PersianCAT with probabilistic accepting policy that uses the first probabilistic accepting policy's set of equations while in PC2 and PC3, PersianCAT uses the second and third equations and PC4 is the implementation of PersianCAT with Quote Beating accepting policy. First column is TSR while second and third columns are profit and Market Share and the last column is the average overall score. The results show that PersianCAT specialists using equations have a better general performance compared to Quote Beating accepting policy.

Table 1. Test runs with GD traders

Agent	TSR	PR	MSH	Mean
PC1	65.146	49.707	56.48	57.111
PC2	57.788	44.888	66.6	56.425
PC3	49.173	37.923	50.19	45.762
PC4	13.087	7.484	26.73	15.767

Table 2. Test runs with ZI-C traders

Agent	TSR	PR	MSH	Mean
PC1	192.228	42.018	45.2	93.148
PC2	191.541	64.804	65.35	107.231
PC3	191.309	92.307	73.87	119.162
PC4	18.34	0.871	15.58	11.597

Table 3. Test runs with Mixed traders

Agent	TSR	PR	MSH	Mean
PC1	193.371	95.067	60.53	351.986
PC2	172.242	17.038	39.05	76.11
PC3	175.903	87.449	80.12	114.49
PC4	8.378	0.444	20.3	3.707

4 Conclusion

In this paper we introduced a new accepting policy to improve the performance of the markets that use double-sided auctions. The result shows that profit, Market share and Transaction Success Rate of the markets can be improved in this way, so more efficiency is reached. In the future we want to improve our specialist by conducting more research in this field so that we can introduce more efficient policies by evaluating effects of evolutionary computation methods and game theory in this field.

References

1. Market Based Control, `http://www.marketbasedcontrol.com`
2. Trading Agent Competition, `http://www.sics.se/tac`
3. Friedman, D., Rust, J.: The Double Auction Market: Institutions, Theories and Evidence, pp. 3–25. Perseus Publishing, Cambridge (1993)
4. Gjerstad, S., Dickhaut, J.: Price Formation in Double Auctions. Games and Economic Behaviour 22, 1–29 (1998)
5. Gode, D.K., Sunder, S.: Allocative Efficiency of Markets with Zero-Intelligence Traders: Market as a Partial Substitute for Individual Rationality. Journal of Political Economy 101(1), 11–137 (1993)
6. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1998)
7. Gerding, E., McBurny, P., Niu, J., Parsons, S., Phelps, S.: Overview of CAT: A Market Design Competition. Version 1.1 (2007)
8. Gerding, E., McBurny, P., Parsons, S.: TAC Market Design: Plannig and Specification. Version 1.03 (2006)
9. MBC Project CAT Tournament Team: CAT Tournament at TAC 2007 Assessment Process

A Framework for Implementing Virtual Collaborative Networks – Case Study on Automobile Components Production Industry

Elham Parvinnia[1], Raouf Khayami[2], and Koorush Ziarati[2]

[1] Islamic Azad University of Mashhad, Mashhad, Iran
[2] Department of Computer Science and Engineering, Shiraz University, Shiraz, IRAN
{parvinn,khayami,ziarati}@shirazu.ac.ir

Abstract. Virtual collaborative networks are composed of small companies which take most advantage from the market opportunity and are able to compete with large companies. So some frameworks have been introduced for implementing this type of collaboration; although none of them has been standardized completely. In this paper we specify some instances that need to be standardized for implementing virtual enterprises. Then, a framework is suggested for implementing virtual collaborative networks. Finally, based on that suggestion, as a case study, we design a virtual collaborative network in automobile components production industry.

Keywords: Virtual collaborative networks, virtual enterprise, Distributed Business Process.

1 Introduction

As a definition virtual enterprise is a temporary network of independent companies that collaborate and integrate their key abilities to respond to a market opportunity. This combination looks like a single unit from outside; and as soon as the request is met or vanished, the reasons for cooperation's existence fade as well, and collaboration will be terminated [1, 2, 3]. The virtual enterprise life cycle can be divided to enterprise configuration, enterprise operation and enterprise dissolution [4].

Before virtual enterprises appeared, we need an information center to save the characteristic of firms that have tendency toward collaboration [5]. These companies who notice their virtual cooperating readiness are called virtual collaboration network.

2 The Necessary Standards for Establishing Virtual Collaborative Networks

"Table 1" summarizes the necessary conditions for establishing virtual collaborative networks.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 909–912, 2008.

Table 1. Necessary conditions for establishing virtual collaborative networks

Necessary conditions for standard	*Purpose and description*
The communication path between members	Independent from any assumption
Exchanging of information and documents	A method without relation to any platforms
Format of information, documents and messages	Select a usual and comprehensive format
Security message transmitting	An available and high secure technique
Definition and modeling of documents	For unified comprehension and facilitate document processing.
Definition of technical dictionary	Integrating the vocabulary of a industry
Definition of companies' core competencies and cooperative conditions	A common structure and format for selecting cooperative members rapidly
A repository for business information and firms' membership	An available repository to facilitate partner selection
Clarifying members of the collaboration	For managing virtual cooperation
Method for business process modeling	Using common, high ability model description
Defining usual business processes for a specific industry	For establish unified definition for business processes
Method of implementation of business processes	A platform independent method with ability of executing distributed business processes
Management of collaboration	A powerful management system for controlling the execution of business processes

An important point in designing a framework for virtual enterprises is the maximum independency of information systems between members. "Table 2" summarizes the items of suggested framework for creating a virtual collaboration network [6, 7, 8].

Table 2. Suggested framework for establishing virtual collaborative networks

Necessary conditions for standard	*Standard*	*Advantages*
The communication path between members	Internet and web	Available and spreading
Documents exchanging	Message passing	Independent from platform
Format of information, documents and messages	XML	Strong structure at document explaining and independent from platform
Security in message transmission	SSL	According to agreement time
Definition and modeling of documents	XML	Ability in describing complex document and flexibility
Definition of technical dictionary	RosettaNet	Homogenous description and unrelated platform
Defining of companies' core competencies and cooperative conditions	XML	Ability in describing complex structures and flexibility
A repository for business information and firms' membership	Website and database	Availability of easy access on homogenous information
Clarifying the members of collaboration	Website and database	Availability of easy and rapid access
Method of business process modeling	UML	The Strongest and the most common tools for description
Defining usual business processes for a specific industry	UML	Describes Ability in distributed business processes
method of implementation of business processes	Web service	Usable in different platforms and capable of XML message sending/receiving
management of collaboration	Website, database and web service	Easy and rapid access to homogenous information

3 Case Study: A Virtual Collaboration Networks in Automobile Components Production Industry

In automobile industry, the manufacturing companies try to do parts of the production out of the factory by ordering parts to other companies. These companies could be dependent on larger automobile companies, or they may work independently. In our project the automobile company has established a firm to supply it with components for the production process. This company acts as virtual collaborative network.

In the automobile production cycle, every collaborating firm based on its role in the virtual enterprise, participates in some business processes. It is possible that one firm accomplishes a role at a time and takes another role in the same business process at another time.

One of the business processes is "primary materials request". In this business process, the part constructor requests primary materials from a supplier. This request should be based on previous agreements and contracts at establishing time of the virtual enterprise. The supplier checks the conformity of the request with the agreement. If the request is correct, the firm will send primary materials to manufacturing firm through a transportation firm. Finally, after the supplier receives the acknowledgement from the manufacturer, the business process will be finished. It is obvious that there are three roles in a "primary materials request" business process: Requester, sender, transporter. We use UML control flow diagram for modeling. In implementation phase, as mentioned before, we use the web service technology. This method has a high ability of exchanging information between two firms with independent

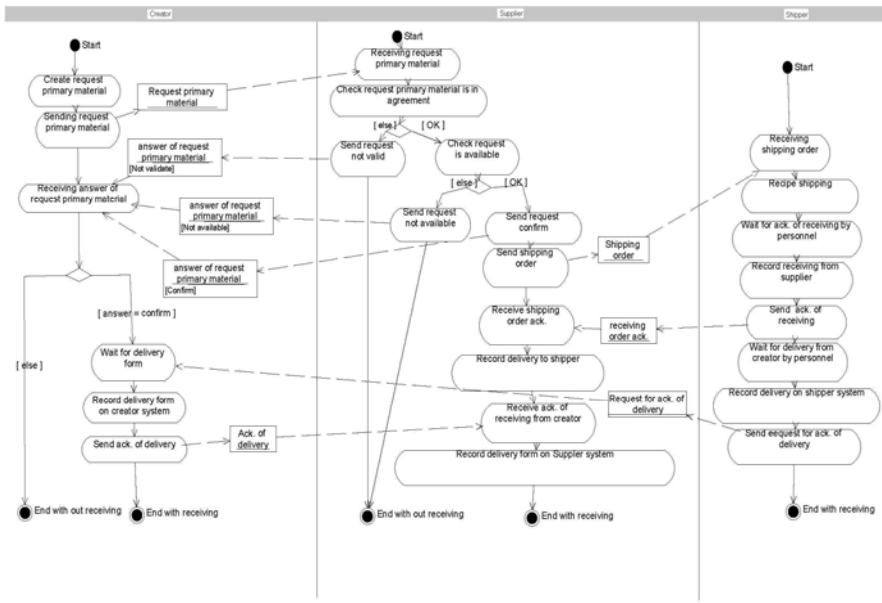


Fig. 1. Business process "primary materials request"

platforms. Regarding the use of SOAP protocol and XML messages in web service technology, we don't need to homogenize the internal systems [9].

4 Conclusion

In this paper, we explored the items that need to be standardized in implementing virtual enterprises. Internet and web have been considered for their environmental connection, because the membership firms are scattered geographically and internet is accessible by all of them. Also, the technologies that work based on XML message passing are suitable for this type of cooperation. Standards like ebXML and Rosetta-Net include models for technical dictionary, core competency of firms, message format, and information format that we considered them to suggest a framework for implementing virtual collaborative networks.

For implementing distributed business processes, we need a technology with which each firm can execute its role in business process independent of its platform. The return values must be sent to partners as a XML message. Web service is an ideal candidate for these requirements. The suggested model in this paper has been used in a virtual collaborative network in automobile part production industry. Some of distributed business processes have been implemented by web service. This implementation confirms the practicality of our framework.

References

1. Franke, U.J.: Managing Virtual Web Organizations in the 21st Century: Issues and Challenges. Idea Group Publishing (2002)
2. Camarinha-Matos, L.M., Afsarmanesh, H.: Collaborative networks: a new scientific discipline. Journal of Intelligent Manufacturing 16, 439–452 (2005)
3. Scholz, ch.: The Virtual corporation: empirical evidences to a three dimensional model. In: Academy of Management, Conference in Toronto (2000)
4. Kim, T., Lee, S., Kim, K., Kim, C.: A modeling framework for agile and interoperable virtual enterprises. Computer in Industry 57, 204–217 (2005)
5. Camarinha-Matos, L.M., Afsarmanesh, H.: Toward a reference model for collaborative networked organizations. In: Proceedings of BASYS 2006, Niagara Falls, Canada, September 4-6, 2006. Springer, Heidelberg (2006)
6. Bussler, C.: B2B protocol standards and their role in semantic b2b integration engines. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering 24(1) (2001)
7. Medjahed, B., Benatallah, B., Bouguettaya, A., Ngu, A., Elmagarmid, A.: Business-to-business interaction: issues and enabling technologies. TheVLDB Journal (2003)
8. Camarinha-Matos, L.M., Afsarmanesh, H.: A modeling framework for collaborative networked organizations. In: Proceedings of PRO-VE 2006, Helsinki, Finland, September 25-27, 2006. Springer, Heidelberg (2006)
9. Estrem, W.A.: An evaluation framework for deploying web services in the next generation manufacturing enterprise. Robotics and Computer Integrated Manufacturing 19, 509–519 (2003)

Virtual Collaboration Readiness Measurement a Case Study in the Automobile Industry

Koorush Ziarati, Raouf Khayami, Elham Parvinnia, and Ghazal Afroozi Milani

Department of Computer Science and Engineering,
Shiraz University, Shiraz, Iran
{ziarati,khayami,parvinn}@shirazu.ac.ir,
afroozi_gh@yahoo.com

Abstract. In end of the last century information and communication technology caused a veritable evolution in the world of business and commerce. Globalization has changed all the commerce equations and business plans. Old companies have to change their strategies if they want to survive after this technological revolution. A new form of collaboration between the distributed and networked organizations has emerged as the "Virtual Organization" paradigm. A company can not join a virtual organization before obtaining a virtual maturity. This maturity shows the readiness of the company to begin a virtual collaboration. In this paper, based on the coherent and formal definition of virtual organizations, the criteria for measuring the readiness of companies are proposed. Our criteria are confirmed, modified or combined by using the factor analysis method on a sufficient number of virtual companies in the automobile manufacturing industry.

Keywords: Virtual Organization, Virtual collaboration, Network Collaboration, Factor Analysis, Virtual Business.

1 Introduction

Undoubtedly business and commerce have been greatly affected by Information and Communication Technology (ICT). Electronic commerce was born and virtual business became a virtual reality. Before that time, the telegram then telephone increased the non face to face trade but, e-mail and internet have increased, exponentially, the virtual trade. In 1986, the term 'Virtual Organization' was introduced by Mowshowtz into the academic discussion [1].

1.1 Basic Definitions

A Virtual Enterprise(VE) is a temporary alliance of enterprises that come together to share skills or core competencies and resources in order to better respond to business opportunities, and whose cooperation is supported by computer networks[2].The other concept that is very similar to the virtual enterprise is the "virtual corporation".

Virtual Organization (VO) has a concept similar to a virtual enterprise, comprising a set of (legally) independent organizations that share resources and skills to achieve its mission / goal, but not only limited to alliance for profit enterprises [3].

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 913–916, 2008.

The VO breeding environment (VBE) represents an association or pool of organizations and their related supporting institutions that have both the potential and the willingness to cooperate with each other through the establishment of a "base" long-term cooperation agreement[4].

2 Virtual Organization Characteristics

Based on the definitions and some case studies, we identify and propose three axes of virtual characteristics as follows[5],[6]:

Core competencies: Core competence as the first axis of virtual characteristics is referred to as "an area of specialized expertise which is the result of harmonizing complex streams of technology and work activity [7].
Cooperation instrument (ICT): Information and Communication Technology (ICT) as the second axis of VC/VE characteristics plays two important roles, first as a solution for integration on a technical level regarding the distributed nature of VC/VE. The second is to keep competitive core competences in the knowledge age, each company having to use different kinds of ICT systems like communication systems and administration systems.
Cooperation culture: In order to participate in virtual corporations, each company requires acting on the basis of some criteria and rules that we call Cooperation Culture. Even though all partners bring along their own cultural profile, in a virtual corporation they have to rely on shared cultural values [6].

Table 1. Three axes and eleven characteristics of virtual collaboration

Core Competencies	Cooperation instrument (ICT)	Cooperation Culture
Specialty	Intra-Organizational Com.	Experience of Cooperation
Customer-based	Inter-Organizational Com.	Flexibility of Management
Separable producing line		Flexibility of Agreements
Productivity		Trust and Assurance
		Transparency

3 Readiness Measurement Criteria

We suggested three axes and eleven characteristics to specify the characteristics of VC/VO/VE to assess company readiness to participate in VWP/VBE. Based on these characteristics, we have designed a questionnaire. The questionnaire includes 70 items which are measured on a 5-point Likert scale.

In general, factor analysis consists of a number of statistical techniques, the aim of which is to simplify complex sets of data which are called factors [8]. Although this solution is in terms of ideal variables, it can be the estimated factor scores- which indicate the amount of virtual characteristics - by using the regression method with the help of SPSS (Statistical Package for Social Science) software.

4 Case Study: Iranian Automobile Industry

As mentioned before, on the basis of VE/VC characteristics, we have suggested some criteria and designed a questionnaire. In order to evaluate the suggested questionnaire (which can be confirmed or not, after omitting some repetitive questions), the automobile industry was selected as a typical industry of the country. Our sample includes 120 corporations which have been selected from the industry index of Iran automobile corporations. Consequently, the 50 filled out questionnaires were gathered from different corporations. This represents a response rate of 42%.

In the factor analysis, we extracted nine factors following the priori criterion derived from our theoretical model, which is also supported by the eigenvalue criterion. 76% of total variance was explained by these nine factors, which is relatively high for social science. This means that we could cover 76% of the variance of all data with nine factors. Other factors have less effect on the modeled data and could be omitted from the analysis, and also the questions that are related to them.

The eleven suggested factors and nine resulting factors from the factor analysis method. We used Varimax rotation and Kaiser Normalization for the factor analysis for finding better relations between questions and factors. We selected factors and questions based on loading factor over 0.5 because these questions and factors had good meaningful relations [9].

As can be seen in Table 2, seven factors of our eleven suggested factors were nearly confirmed by the factor analysis results.

The reliability analysis showed internal consistency of questions that were allocated to factors. The overall alpha coefficient of our questions allocating to our model was 0.88, which suggests a high reliability.

Table 2. The Relation of the Factors

SUGGESTED FACTORS	Questions of suggested factors	RESULTED FACTORS	Questions of resulted factors
Trust and Assurance	Questions 55... 64	Trust and Assurance	Questions 56, 57, 59, 60, 61, 62
Intra-Organizational Com(ICT)	Questions 24..34	Intra-Organizational Com. (ICT)	Questions 24, 25, 28, 29, 31..34
Transparency	Questions 65..70	Transparency	Questions 18, 66, 69, 70
Flexibility of Management	Questions 46..51	Flexibility of Management	Questions 45, 46, 47, 51, 55, 63
Inter-Organizational Com. (ICT)	Questions 35..42	Inter-Organizational Com. (ICT)	Questions 27, 30, 35, 38, 39, 42, 44
Separable Producing Line	Questions 15, 16	Separable Producing Line	Questions 15, 16
Specialty	Questions 1..9	Specialty	Questions 1..4, 7, 8, 12, 19, 22, 26
Customer-Based	Questions 10..14	Customer-Based	Questions 5, 6, 10, 11, 17, 20, 21, 23, 40
Productivity	Questions 17..23		
Experience of Cooperation	Questions 43..45	Experience and Interested in Cooperation	Questions 37, 43, 50, 52, 53, 54, 58, 64
Flexibility of Agreements	Questions 51..55		

5 Conclusion

In this paper, based on the common definition of a virtual company, some important characteristics for measuring the readiness of companies to join a virtual network cooperation are proposed. Based on these characteristics, a questionnaire with seventy questions was prepared for elite companies in the automobile industry. These companies have been selected based on virtualness characteristics as the ideal virtual company. For these companies virtual activities had already begun. Most of these questions could be redundant or focused on the same characteristic and the same fact. So, to check our suggested characteristic and subsequently, to obtain the proper instrument for comparing companies, we used factor analysis.

After applying the factor analysis method by using SPSS software, 9 important factors are obtained. These results confirm a part of our suggested characteristics and complete it. Knowing how to use these factors, one can easily determine the degree of virtuality of the other companies in the same industry. The contribution of this paper is to propose a systematic way of measuring the readiness of companies that want to begin network collaboration.

It seems that these criteria could be different from one country to another and from one industry to another. But, with the business and market globalization phenomena, we think that common criteria could emerge in the near future.

References

1. Franke, U.J.: The concept of virtual web organizations and its implications on changing market conditions. Electronic Journal of Organizational Virtualness (2001)
2. Camarinha-Matos, L.M., Afsarmanesh, H.: The virtual enterprise concept, in infrastructures for virtual enterprises – networking industrial enterprises, vol. 153. Kluwer Academic Publishers, IFIP (1999)
3. Camarinha-Matos, L.M., Afsarmanesh, H.: Collaborative Networked Organizations. Kluwer Academic Publishers, Dordrecht (2004)
4. Camarinha-Matos, L.M., Afsarmanesh, H.: Elements of a base VE infrastructure. J. Computers in Industry 51, 139–163 (2003)
5. Bauer, R., Köszegi, S.T.: Measuring the degree of virtualization. Electronic Journal of Organizational Virtualness (2003)
6. Scholz, Ch.: The virtual corporation: empirical evidences to a three dimensional model. In: Conference of Management, Toronto (2000)
7. Hamel, G., Prahalad, C.K.: The core competence of the corporation. Harvard Business Review 68, 79–91 (2003)
8. Kline, P.: An Easy Guide to Factor Analysis, Routledge (1994)
9. Johnson, R.A., Wichern, D.W.: Applied Multivariate Statistical Analysis, 5th edn. Prentice-Hall, Englewood Cliffs (2002)

Facilitating XML Query Processing Via Execution Plan

Sayyed Kamyar Izadi, Vahid Garakani, and Mostafa S. Haghjoo

Department of Computer Engineering,
Iran University of Science and Technology, Tehran, Iran
izadi@iust.ac.ir, vahidgarakani@comp.iust.ac.ir,
haghjoom@iust.ac.ir

Abstract. XML queries are based on path expressions which are composed of some elements connected to each other in a tree pattern structure, called Query Tree Pattern (QTP). Thus, the core operation of XML query processing is finding all instances of QTP in the XML document. A number of methods are offered for QTP matching, but they process too many elements in XML document while most of them have no opportunity to participate in the final result. In this paper we propose a novel method which doesn't blindly processes elements of the document. We use a query execution plan called *ResultGraph* generated by execution of path expression on the abstraction of the XML document called *QueryGuide*. In contrast to the existing methods, in our method only elements in the document which have the same tag name as the leaves of QTP are processed. Also we only process elements which have a chance to produce a result and those which are definitely not part of any final result are ignored. Experimental results show the efficiency of our method.

Keywords: XML Query Processing, Query Tree Pattern, Dewey ID, Twig Join, Twig Matching, QueryGuide, ResultGraph.

1 Introduction

Query processing is an essential part of any type of databases as well as XML databases. XML query processing is much more complicated than traditional query processing methods because of the structure of XML. A path expression specifies patterns of selection predicates on multiple elements related by a tree structure named *Query Tree Pattern* (QTP). Consequently, In order to process an XML query, all occurrences of its related QTP should be distinguished in the XML document. This is an expensive task when huge XML documents are attended. A number of methods are proposed to answer this type of queries and we have classified these methods into three groups. The first group contains methods which are based on *structural join* [1][4] which produce large amount of intermediate results. Second group contains methods which process all the elements in the XML document having the same tag name with one of the query nodes [2][3] and the third group contains methods which only process elements which belong to the leaves of QTP in the XML document [6].

None of the above query processing methods uses indexes or summaries efficiently. We set two goals in this paper; first we want to process only elements which belong to the leaves of QTP in the XML document and second to process only

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 917–920, 2008.

elements which have a chance to participate in the final result and ignoring those which are definitely not part of any result.

2 The Proposed Method

Our proposed method, called RG, has two main steps. In the first step, query is executed over the *QueryGuide* of the document to produce *ResultGraph*. In the second step, final result is constructed by applying *ResultGraph* on the XML document.

2.1 QueryGuide

QueryGuide acts as a guide for query processing to prevent blind processing of nodes in the XML document. It is used to create an execution plan before executing of query directly on the XML document to minimize the number of comparisons.

Our Q*ueryGuide* is inspired by *DataGuide* [5] which is summary of semi-structured documents. But our goal of employing Q*ueryGuide* is more than accessing nodes faster. Structural summaries or path indexes contain an abstraction of XML documents which can help complicated queries with more than one branch to be evaluated more efficiently.

QueryGuide is a tree structure which is made over an XML document labeled by Dewey method. Each node in *QueryGuide* is a representative of all elements in the XML document which have the same path. Each element in the XML document is associated to only one node in the *QueryGuide*. The path string of an element in the XML document is equal to the path string of its associated node in the *QueryGuide*.

2.2 ResultGraph

ResultGraph acts as an XML query execution plan which is introduced in our proposed method to prevent the query processor from searching the XML document blindly. *ResultGraph* is a labeled graph defined by $RG = (V, E, L)$ where V is a set of *QueryGuide's* nodes with the same name as QTP's leaves, $E \subseteq V \times Z+ \times V$ is ternary relation describing the edges (including the labeling of the edges) and $L: E \rightarrow Z+$ is a function, maps each edge to its label.

An example of *ResultGraph* construction could clarify this structure more. Consider the sample *QueryGuide* in Fig. 1(a) and the sample QTP in Fig. 1(b). It is needed to execute the QTP over the *QueryGuide* in the first step. Here *B* and *D* are the leaves of the QTP. Therefore, we will find all occurrences of these nodes in the *QueryGuide* which have a chance to participate in the final result. The first nodes which we will examine are D_1 and B_1. It is clear from *QueryGuide* that D_1 is the child of C_2 and C_2 is the child of A_2, thus D_1 matches the right branch of the QTP. At the other hand *B1* is the child of A_2 and matches the left branch of the QTP. Hence, A_2, C_2, B_1, and D_1 match the QTP altogether. Since *B* and *D* are the leaves of the QTP, an edge with vertexes B_1 and D_1 is added to the *ResultGraph*. Label of the edge will be set *2* because two branches of the QTP are joined to each other in the second level of QTP. There is the same story for other edges in the *ResultGraph*. The algorithm for producing the *ResultGraph* of a QTP with two leaves is as the following:

1. Choose all nodes of *QueryGuide* with the same name as QTP leaves.
2. Find all combination of nodes that match the QTP
3. For each combination do:
    a. Find the NCA of the combination.
    b. Navigate from NCA to the root of *QueryGuide*. For all of the nodes with the same type of QTP's root add an edge for the combination to the *ResultGraph* with the label equal to the level of the traversed node in the *QueryGuide*.

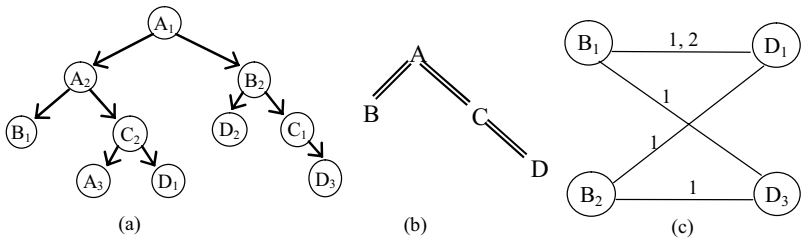


Fig. 1. (a) A sample QueryGuide (b) A sample QTP (c) Related ResultGraph

2.3 Matching Process

Final result is constructed based on the *ResultGraph* in the matching process. Each edge of the *ResultGraph* guides the matching process to produce a fragment of the final result. Therefore, final result is the union of partial results produced for each edge of the *ResultGraph*. Now consider a given edge in the *ResultGraph* and its vertexes. Each vertex is a node in the *QueryGuide*. As mentioned before, each node in *QueryGuide* has an associated ordered list of pointers. In order to produce part of final result, these two lists (for example L_1 and L_2) should be compared to be able to find nodes which their Dewey label match up to the level introduced by the label of the edge. The equality relation between two Dewey labels is considered for the prefixes of Dewy labels up to the l_{th} level of their Dewey labels which l is the label of the relevant edge.

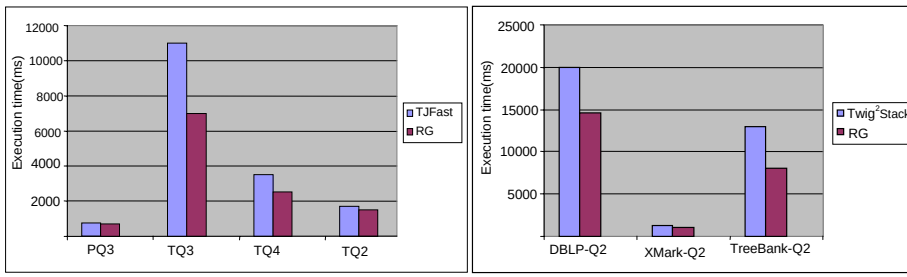
Current nodes of the two lists are set to the first member of each list. The current node of L_1 is compared to the current node of L_2. If the prefixes of Dewey labels of the current nodes of L_1 and L_2 are not the same then the current node of the list with smaller label is shifted to the next node. This is done while nodes with equal prefixes of Dewey labels are reached. Now a matching is occurred and all the nodes in the two lists which have the same prefix as the selected nodes are chosen. Each combination of these nodes from the two lists will be a match for the final result. The process of matching will be continued till the end of one of the two lists is reached. The label of inner nodes of QTP is easily produced regards to the *ResultGraph* edge and the *QueryGuide*.

As an example, consider edge B1_D1 (the label of the edge is assumed 2) of Fig. 1(c) with sample lists:

B1 = {1/3/1, 2/5/1, 2/7/1}
D1 = {1/2/2/1, 1/3/3/1, 1/3/3/2, 2/6/2/2, 2/7/1/2}
Matching these two lists will be resulted in the following result list:
Result = {(1/3/1, 1/3/3/1), (1/3/1, 1/3/3/2), (2/7/1, 2/7/1/2)}.

3 Experimental Results

In this section because of limitation of space we present only most two important experiments. We compare our proposed method with *Twig²Stack* [3] and *TJFast* [6]. Our experiments run on a PC with 2.2 GHz Intel Pentium IV processor running Red Hat Linux 8.0 with 2 GB of main memory. We use the same queries and datasets which are used in [3] and [6]. Source of queries and information about dataset could be found in these references.



(a) (b)

Fig. 2. (a) TJFast versus RG (b) Twig²Stack versus RG

References

1. Al-Khalifa, S., Jagadish, H.V., Koudas, N., Patel, J.M., Srivastava, D., Wu, Y.: Structural Joins: A Primitive for Efficient XML Query Pattern Matching. In: ICDE Conference, pp. 141–152 (2002)
2. Bruno, N., Koudas, N., Srivastava, D.: Holistic Twig Joins: Optimal XML Pattern Matching. In: SIGMOD Conference, pp. 310–321 (2002)
3. Chen, S., HuaGang, L., Tatemura, J., Hsiung, W., Agrawal, D., Candan, K.S.: Twig²Stack: Bottom-up Processing of Generalized-Tree-Pattern Queries over XML Documents. In: VLDB Conference, pp. 283–294 (2006)
4. Chien, S., et al.: Efficient Structural Joins on Indexed XML. In: VLDB Conference, pp. 263–274 (2002)
5. Goldman, R., Widom, J.: DataGuides: Enabling Query Formulation and Optimization in Semistructured Databases. In: VLDB Conference, pp. 436–445 (1997)
6. Lu, J., Ling, T.W., Chan, C.Y., Chen, T.: From Region Encoding to Extended Dewey: On efficient processing of XML twig pattern matching. In: VLDB Conference, pp. 193–204 (2005)

The Impact of Hidden Terminal on WMNet Performance

Kourosh Hassanli[1], Ali Khayatzadeh Mahani[2,3], and Majid Naderi[3]

[1] Department of Telecommunication Engineering,
Islamic Azad University-Jahrom Branch,
Jahrom, Iran
[2] Faculty of Engineering & Technology,
Shahid Bahonar University of Kerman,
Kerman, Iran
[3] Department of Electrical and Electronic Engineering,
Iran University of Science and Technology, Narmak,
Tehran, Iran
{k_hassanli,khayatzadeh,m_naderi}@iust.ac.ir

Abstract. Hidden terminal is one of the major effective parameters on wireless networks performance. In order to analyze wireless networks exactly on a mathematical basis, the hidden terminal effect is also needed to investigate. In this paper, it is tried to propose the mathematical model of DCF mode of IEEE 802.11 standard with regarding the hidden terminal effect, and then quantify its impact on network performance. So the M/G/1 queue is used to calculate packet delay and network throughput, in which the collision probability is the main factor of the packet service time, and the collision probability of each packet has been considered as a function of three quantities: the number of neighbors, the neighbors of destination and the weighted number of nodes in second hop neighborhood. After modeling and numerical analysis, the results have been compared with the similar model irrespective of the hidden terminal effect.

Keywords: Hidden Terminal, Queuing Theory, DCF, IEEE 802.11.

1 Introduction

In recent years, configurable wireless networks, especially Ad-hoc networks, have been well-considered in wide range applications such as military, commercial and academic applications [1]. In order to implement an effective routing algorithm, admissions control or topology control, in MANet, it is necessary to analyze the network performance and identify their shortcomings and limitations accurately. Because of complicated interactions between layers of wireless networks, especially in multi-hop networks, and lack of a comprehensive mathematical model, it is very important to analyze their limitations fundamentally [2, 3].

Hidden terminal is one of the fundamental problems in wireless networks that impress the network performance considerably [4]. So the mathematical model of IEEE 802.11 standard with regarding hidden terminals would be help us to modify the IEEE 802.11 standard for other applications e. g. wireless mesh networks[5].

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 921–925, 2008.

The rest of the paper is organized as follows: In section 2 the hidden terminal effect is quantify. The model is analyzed numerically in section 3. Finally the paper is concluded in section 4.

2 Proposed Method

Proposed model quantify the hidden terminal effect for multi-hop wireless ad hoc networks and then use the Mcdonald model [2] as basis to develop the mathematical analysis of performance evaluation.

2.1 Hidden Terminal Effect on Multi-hop Networks

Hidden terminal is one of the major effective parameters on wireless networks performance. In order to describe it, three levels of interference area are defined for each node in a wireless network [4].

Level-1: including all the nodes which are neighbor of source and destination (N_1). Level-2: including all the nodes which are neighbor of destination only (N_2). Level-3: including all the nodes which are two-hop neighbors of destination only.

To analyze the hidden terminal effect, we use the weighted two-hop neighbors in level-3 (N_3). So we can list the effect of these levels on network performance as follows:

1- All transmissions originated by nodes in level-1 and level-2 interference set will be deferred by the ongoing communication.

2- Only the transmission originated by nodes in level-3 interference set toward nodes in level-2 interference set will be deferred by the ongoing communication.

Therefore, not only the neighbors but also the nodes in the second hops take effect on service time of queued packets [4]. Now it is possible to write the equation of collision probability as follows:

$$p = 1 - (1-\tau)^{N_1}(1-2R_{hidden}\tau)^{N_2+N_3} \quad (1)$$

In which R_{hidden} is the time interval from beginning to transmit RTS signal till beginning to transmit CTS signal. R_{hidden} is doubled, because hidden terminals may have transmitted a colliding packet in a slot before or after R_{hidden}.

2.2 Collision Probability Considering Hidden Terminals

The following assumptions help us in quantitative analysis of the hidden terminal effect:

1) The number of neighbors is equal to $n-1$.2) Destination node is selected randomly among neighbors of source node, considering a uniform traffic pattern.3) the nodes are distributed in the network area uniformly.4) the interference area of each node is circular with radius R.

As the traffic pattern is assumed uniform and the nodes are distributed uniformly, N_2 is calculated by:

$$N_2 = \frac{A_{level-2} \cdot n}{\pi R^2} \tag{2}$$

In which $A_{level-2}$ is the area of level-2 as shown in figure 1-left. In order to simplify the relations for calculating N_3, the circles C_1 and C_2 are replaced by another circle C' as shown in figure 1-right.

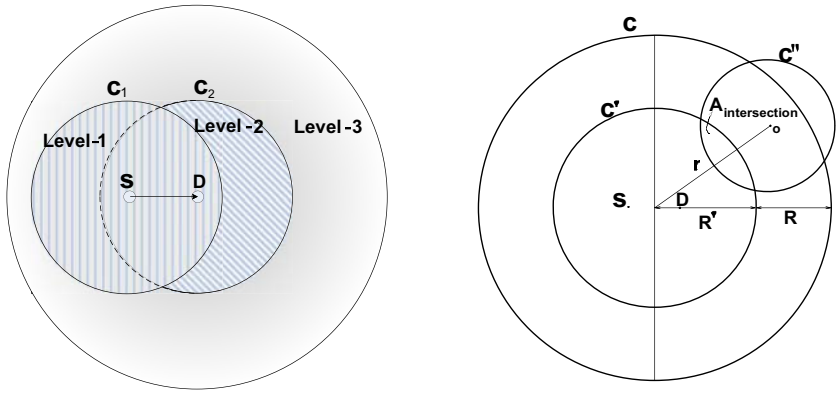


Fig. 1. Left-The source-destination communication area and their tripartite level of interference Right- the intersection area to calculate the weighted two-hop neighbors

In fact, N_3 is the weighted two-hop neighbor located in level-3. So it represents the fraction of traffic due to nodes in level-3 that ends to nodes in level-2. Thus N_3 is calculated as follows:

$$N_3 = \int_{R'}^{R'+R} \frac{A_{intersection}}{\pi R^2} \cdot \frac{2\pi r}{A_{ring}} dr \tag{3}$$

where A_{ring} represents the area between two outer and inner circles and $A_{intersection}$ can be calculated as follows:

$$A_{intersection} = R^2 \cos^{-1}\left(\frac{r^2 + R^2 - R'^2}{2rR}\right) + R'^2 \cos^{-1}\left(\frac{r^2 + R'^2 - R^2}{2rR'}\right) - \sqrt{(-r+R+R')(r+R-R')(r-R+R')(r+R+R')} \tag{4}$$

Now we can manipulate the collision probability, equation 2, by use of N_2 and N_3 values. So the packet transmission delay is obtained by substituting (2) and (3) in the proposed queuing model in [4].

3 Numerical Results

We derived some performance through a proposed model. We considered a WLAN Mesh network whose nodes are uniformly distributed. Three scenarios are studied: 2,

6 and 14 neighbors for each node. The packet size is set to 1024 bytes, and Negotiation messages as well as physical headers are transmitted at 1 Mbit/s, while data and acknowledgments are sent at 11 Mbit/s.

The delay of DCF in presence of hidden terminal is compared against that without hidden terminal.

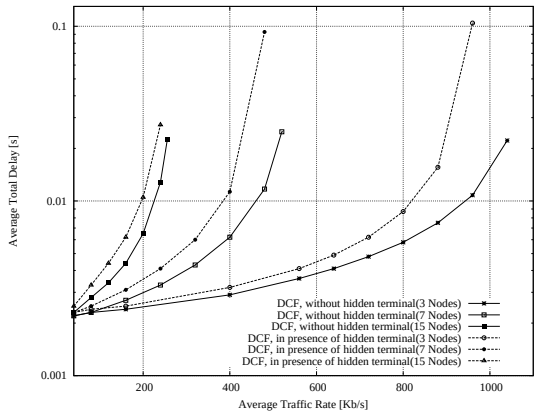


Fig. 2. The average total delay vs. traffic load per node, in presence of hidden terminal and without hidden terminal. Results are shown for different values of the number of nodes in each other's radio proximity.

Figures 2 shows the delay of the whole network. The results are shown for different values of the average nodes neighborhood size, i.e., average number of nodes in radio proximity of each other.

The results have been obtained as the average offered traffic load per node varies, and for different values of the number of nodes in the network, i.e., under different traffic load conditions. Looking at the plot, we note that, the DCF total transmission delay without hidden terminal is lower than DCF in presence of hidden terminal.
The packet delay is computed as the period between the packet generation and its successful delivery to the destination. Again, results are presented for different values of the number of nodes. So if the effect of hidden terminal is ignored a big gap would be formed between average packet transmission delay in presence of hidden terminal and without hidden terminal.

4 Conclusion

This paper reviews some analytical model of a DCF based ad hoc networks. The main contribution of the paper is MAC analytical model for multi-hop wireless networks with regarding the hidden terminal effect. The analytical results are extracted for more various traffic conditions. The results show the different between packet transmission delay in presence of hidden terminal and without hidden terminal. So the proposed

model would be used in call admission control mechanism which is essential for DCF in real time applications.

Our future work has been initiated to introduce a MAC protocol to decrease the impact of hidden terminal effect in multi-hop wireless networks.

Acknowledgment. The authors would like to thank Iran Telecommunication Research Center for their all-inclusive support service on this work.

References

[1] Cheng, S.T., Wu, M.: Performance Evaluation of Ad-Hoc WLAN by M/G/1 Queuing Model. In: Proceeding of the international conference on information technology: Coding and Computing, ITCC 2005, vol. 2, pp. 681–686 (2005)

[2] Ozdemir, M., Mcdonald, A.B.: On the performance of ad hoc wireless LANs: a practical queuing theoretic model. J. performance evaluation 63(11) (2006)

[3] Bianchi, G.: Performance Analysis of the IEEE 802.11 Distributed Coordination Function. J. Selected Areas in Communications 18, 535–547 (2000)

[4] Ozdemir, M., Mcdonald, A.B.: A Queuing Model of Multi-hop Wireless Ad hoc Networks with Hidden nodes. In: 7th IFIP International Conference on Mobile and Wireless Communications Networks, MWCN 2005, Morocco (2005)

[5] Mahani, A.K., Naderi, M., Chiasserini, C.F., Casetti, C.E.: Enhancing Channel Utilization in Mesh Networks. In: Proceeding of MILCOM 2007. IEEE Press, Los Alamitos (2007)

[6] Ray, S., Starobinski, D., Carruthers, J.B.: Performance of wireless networks with hidden nodes: a queuing-Theoritic analysis. J. Computer Communications 28, 1179–1192 (2005)

[7] Zahedi, A., Pahlavan, K.: Natural hidden terminal and the Performance of the Wireless LANs. Universal Personal Communications Record (1997)

[8] Wireless LAN Mediume Access Control (MAC) and physical layer (PHY) specification: Higher speed physical layer (PHY) extension in the 2.4GHz band, IEEE std. 802.11b/D5.0 (1999)

Design and Implementation of a Fuzzy Accident Detector

Shahram Jafari[1], Mohammad Arabnejad[2], and Ali Rashidi Moakhar[2]

[1] Dept. of Computer Systems and Engineering, Shiraz University, Iran
[2] Dept. of Electronics Engineering, Islamic Azad University of Kerman, Iran
Jafaris@shirazu.ac.ir, shj53@yahoo.com

Abstract. A fuzzy accident detector has been proposed in this paper. The implemented controller ensures a reliable margin for the speed of a car. This is done by carefully observing the skills of the driver in controlling the automobile during a critical condition. Since *x*- and *y*- accelerations of the automobile change sharply during an accident, such conditions can be detected. The system also updates the speed limits in different locations on the road.

Keywords: Fuzzy accident detector, Neural Network, Analog to Digital Converter (ADC), Multi-Media Card (MMC).

1 Introduction

New generations of cars are improved in such a way that the number of accidents decreases. Innovative ideas have emerged and implemented in order to reduce the risk of car accidents. During the past recent years, some alarm systems and intelligent control apparatus have been designed and developed in order to increase the safety of automobiles.

For instance, the car-to-car anti-collision system [10] uses vehicle to vehicle 5.8-GHz frequency band communication and gives a proper signal to the driver before an accident occurs. The system is well verified, however, it is unable to act mechanically in order to stop the car from an accident. It does not consider a limitation for the speed of the car and it needs some extra apparatus for the communication purposes between the automobiles. In [10], communication is dependent to the information provided by other cars which are equipped with the same apparatus. Due to the lack of communication between the car which is not equipped with such a gadget and other cars, the driver of the car with communication capability may be mislead or at least he may underestimate the potential risk of accident.

Lack of evaluation mechanism for the driver of the car is clearly observed in all of the implementations [11-16]. Also, the ability of the driver in controlling dangerous and unpredictable situations is not tested as a critical parameter.

In other words, speed control of the vehicle only relies on the capabilities of the vehicle instead of skills of the driver. By increasing the flexibility of the safety system, this problem is overcome in our implementation (Fig. 1).

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 926–929, 2008.

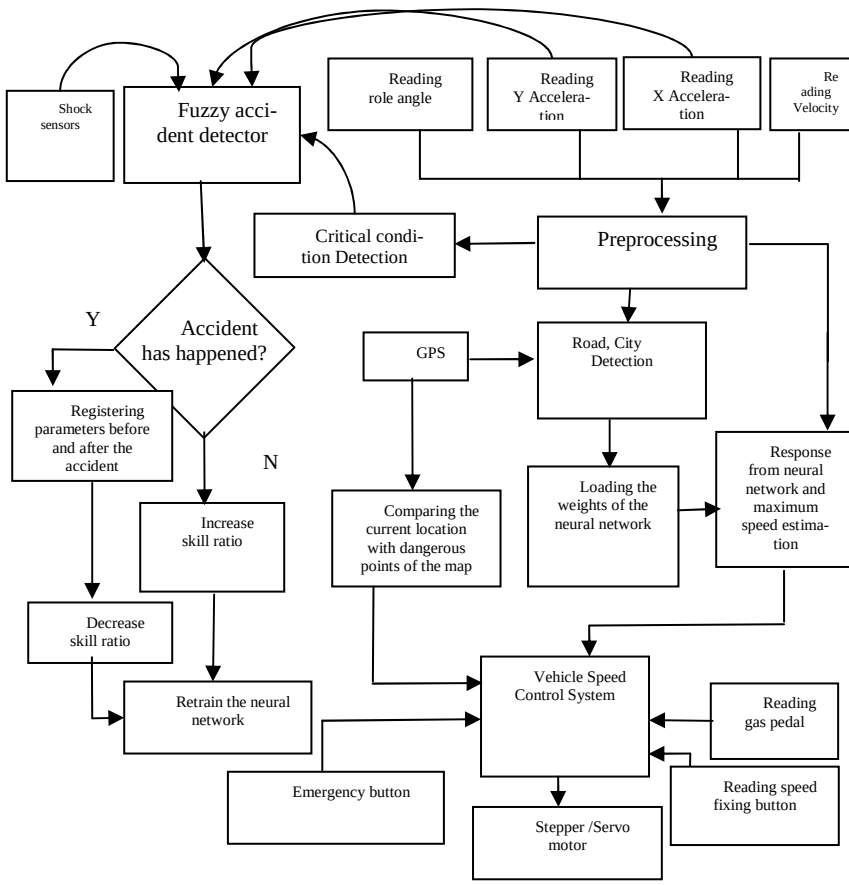


Fig. 1. General diagram of the implemented System

2 Implemented Fuzzy Accident Detector

During critical conditions *x*- and *y*- accelerations of the automobile change sharply. This information together with the output of shock sensors implemented in the system help to detect an accident. A collision is not a clearly defined phenomenon also the level of impact has a reverse proportion with the skill of the driver. Therefore, a measure of impact must be applied to the retraining phase of the neural network (Fig. 1) and an accident can be defined as a fuzzy term. Retraining of the neural network is necessary to convey the driver's skill and it is partly done by observing how severe the accident is. Changes of *x*- and *y*- accelerations of the automobile are defined as fuzzy variables. These fuzzy quantities are, then, utilized in the fuzzy rules so that the outcome will show whether an accident has occurred and how severe it is. Even if the output of this part of fuzzy inference presents a slight collision, it is accumulated with

the result of shock sensors (using AND operator) to provide the final output. This is done in order to make a stronger decision whether an accident has occurred or not.

Membership diagram for the *x*-acceleration of the automobile which has been divided into 4 fuzzy sets has been shown in Fig. 2. Also, one of the fuzzy inference rules has been stated as follows:

Fuzzy Rule$_1$ ***(Accident occurred)***

IF ***in t_1:* a *(x- acceleration) is high-negative***
And ***in t:* a *is positive***
And ***shock sensors subsystems show accident***
Then ***Accident has occurred.***

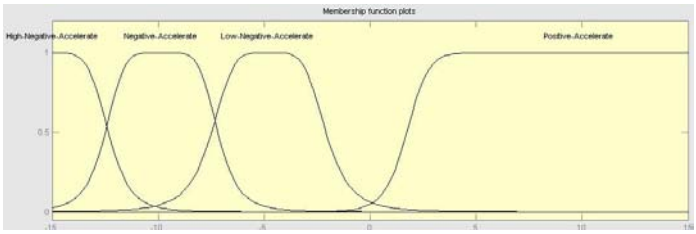


Fig. 2. Membership function for fuzzy variable *x- acceleration* of the automobile

3 Discussion

After running many real-time experiments, it was observed that the implemented system unlike other anti-collision car safety systems adapt with both the car and the driver to control the speed of the car. A fuzzy accident detector cooperates with other decision making modules to control the speed of car by ultimately controlling the amount of gas which is pumped into the engine. After practically testing the controller, it was verified that such systems help to reduce the risk of car road accidents.

Acknowledgments. The authors would like to thank Mr. Mehdi Sharifi and Mr. Mohammad ali Nikouei for all their efforts during this part of the project.

References

1. Handmann, U., Leefken, I., Tzomakas, C., et al.: A flexible architecture for intelligent cruise control. In: Proceedings of the IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems, 1999, pp. 958–963 (1999)
2. Jafari, S., Shabaninia, F., Nava, P.A.: Neural network algorithms for tuning of fuzzy certainty factor expert systems. In: IEEE/WAC World Automation Congress, USA (2002)
3. Jafari, S., Jarvis, R.A.: Robotic Hand Eye Coordination: From Observation to Manipulation. Int. Journal of Hybrid Intelligent Systems (IJHIS) (September 2005) (special issue)
4. SAMSUNG Multimedia Card Product Datasheet (February 2005)
5. Datasheet of microcontrollers (May 10, 2006), `http://www.alldatasheet.com`

6. GPS information (June 10, 2005), http://www.garmin.com
7. AVR/ Atmel microcontroller datasheet (May 10, 2005), http://www.atmel.com/avr
8. http://www.mcselec.com (May 10, 2005)
9. Jones, J.: Networks, 2nd edn. (May 10, 1991), http://www.atm.com
10. Dankers, A., Fletcher, L., Petersson, L., Zelinsky, A.: Driver Assistance: Contemporary Road Safety (May 10, 2005), http://www.araa.asn.au/acra/acra2003/papers/21.pdf
11. http://citeseer.nj.nec.com/cache/papers/cs/767/ http:zSzzSzrobotics.eecs.berkeley.eduzSz~ godbolezSzitsframework.pdf/framework-for-the-analysis.pdf (May 10, 2005)
12. A simulator-based investigation of visual, auditory, and mixed-modality display of vehicle dynamic state information to commercial motor vehicle operators, Steven Mark Belz (May 10, 2005), http://scholar.lib.vt.edu/theses/available/etd-11297-195327/unrestricted/thesis.pdf
13. Investigation of Alternative Displays for Side Collision Avoidance Systems, NHTSA (May 10, 2005), http://www.itsdocs.fhwa.dot.gov/jpodocs/repts_te/41n01.pdf
14. Gustafsson, F., Jansson, J.: Decision Making for Collision Avoidance Systems (May 10, 2005), http://www.control.isy.liu.se/~fredrik/reports/02SAEfca.pdf
15. Basic collision warning and drive information systems: Human factors research needs (May 10, 2005), http://www.tfhrc.gov/humanfac/98-184.pdf
16. Preliminary Human factors for Intelligent vehicle initiative (May 10, 2005), http://www.itsdocs.fhwa.dot.gov/edldocs/7363/ch03/ ch03_02.html
17. Jang, J.-S.R.: ANFIS: adaptive-network-based fuzzy inference system. IEEE Transactions on Systems, Man and Cybernetics 23(3), 665–685 (1993)
18. Larson, R., Keckler, W.: Optimum adaptive control in an unknown environment. IEEE Transactions on Automatic Control 13(4), 438–439 (1968)
19. Negnevitsky, M.: Artificial Intelligence: A Guide to Intelligent Systems 1/e, p. 394. Addison Wesley, Reading (2002)
20. Paola, J.D., Schowenderdt, R.A.: A review and analysis of backpropagation neural networks for classification of remotely sensed multispectral imagery. International Journal of Remote Sensing 16, 3033–3058 (1995)

Approximation Algorithms for Edge-Covering Problem

Mohammad Hosseinzadeh Moghaddam[1] and Alireza Bagheri[2]

[1] Computer Engineering Department,
Islamic Azad University of Hashtroud ,Hashtroud, Iran
[2] Computer Engineering and IT Department,
Amirkabir University of Technology, Tehran, Iran
mh.moghaddam@yahoo.com,
ar_bagheri@aut.ac.ir

Abstract. In edge-covering problem the goal is finding the minimum number of guards to cover the edges of a simple polygon. This problem is *NP*-hard, and to our knowledge there is just one approximation algorithm for a restricted case of this problem. In this paper we present two approximation algorithms for two versions of this problem.

Keywords: Art gallery, edge-covering, approximation algorithm, vertex guard, edge guard.

1 Introduction

In 1973 Victor Klee posed the following question: How many guards are necessary, and how many are sufficient to patrol the paintings and works of the art in an art gallery with n walls? This question initiated many researches on the various versions of the problem, and many papers and even books have been published on this problem, see [1, 2, and 3] for comprehensive surveys. In some versions of the art gallery problem, one should find the minimum number of guards to patrol interior of a polygon. This means that every point inside the polygon is visible from at least one guard. Point p is visible from point q, if line segment $\overline{pq}$ lies completely inside the polygon. Guards may be placed inside or on the boundary of the polygon, and may be stationary or mobile. If stationary guards are restricted to be placed on the vertices of the polygon, they are called *vertex guards*; otherwise they are called *point guards*. Mobile guards that are restricted to move only along the edges of the polygon are called *edge guards*. A point is visible from an edge guard, if it is visible from at least one point of the edge on which the guard moves.

Laurentini [4] investigated a version of the art gallery problem, in which the guards should patrol the walls of the art gallery instead of patrolling its interior. This version was called *edge-covering*, and was compared to interior cover. He proved that the edge-covering problem is *NP*-hard. He introduced a restricted version of this problem, in which an edge should be completely visible from at least one point guard. He showed that this version is also *NP*-hard, and proposed an approximation algorithm with a logarithmic factor for that. In this paper we propose two approximation algorithms with a logarithmic factor for vertex guard and edge guard versions of the edge-covering problem for simple polygons.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 930–933, 2008.

2 Approximation Algorithms

Let $VP(P, z)$ denote the set of all points of P that are visible from a point $z \in P$. We call $VP(P, z)$ *the visibility polygon* of P from z. A point $z \in P$ is said to be *weakly visible* from an edge e of P if there exists a point u on e such that the segment $\overline{zu}$ lies inside P. Let $VP(P, e)$ denote the set of all points of P that are weakly visible from e. We call $VP(P, e)$ *the weak visibility polygon* of P from e.

First, we consider the vertex guard version of the problem and present our approximation algorithm, and then we extend the proposed algorithm for the edge guard version. In the vertex guard version of the edge-covering problem, we should find the minimum number of vertex guards to patrol the edges of the given polygon. Passing lines through every pair of the vertices of the polygon we divide the edges of the polygon into some segments, are called *the basic edge segments.*

Definition 1. Edge segment l_1 on edge e is called *basic edge segment*, if there is no other edge segment l_2 such that $l_1 \subset l_2$ and it can be divided by a line passing through the two vertices of the polygon.

Lemma 1. Every basic edge segment is completely visible from a vertex of the polygon.

Proof. Let us assume on the contrary that there is a basic edge segment l that is partially visible from vertex v_j. Thus there is another vertex v_i that line $\overline{v_i v_j}$ divides l, a contradiction. □

Thus, the vertex guard version of the edge-covering problem can be considered as an instance of the set covering problem [5]. The set that should be covered is the set of the basic edge segments, and the subsets are the sets of the basic edge segments that are visible from the vertices of the polygon. To find the basic edge segments, we compute the visibility polygons of the vertices, and find the intersections of their boundaries. Then we use Johnson's approximation algorithm for the set covering problem [6]. Since Johnson's algorithm has a logarithmic approximation factor, our algorithm also has. The algorithm is given by Algorithm 1.

In the following we compute the time complexity of the algorithm. In step 1, computing the visibility polygon of each vertex of the given simple polygon takes $O(n)$ time [7]. Each visibility polygon consists of two types of edges, the edges that are on the boundary of the original polygon and the constructed edges that lie inside the original polygon. The basic edge segments are on the boundaries of the visibility polygons, and can be computed by finding the intersections of all the edges of the visibility polygons. Each visibility polygon has at most $O(n)$ edges on its boundary and there are n visibility polygons. Intersections of the edges of the two visibility polygons with $O(n_1)$ and $O(n_2)$ edges can be found in $O(n_1 + n_2)$ time, and in the same time the edges of the two visibility polygons can be updated. This is due to this fact that the edges are segments of the edges of the original given simple polygon and are ordered according to its boundary. The union of two visibility polygons with $O(n_1)$ and $O(n_2)$ edges has $O(n_1 + n_2)$ edges. Hence, the intersections of all the edges

of all the visibility polygons can be computed in $O(n^2)$ time using a simple divide and conquer approach.

In step 2, we need to have all F_j for all $1 \le j \le n$, which are the set of all the basic edge segments that are completely visible from v_j. These sets have already obtained in step 1. Steps 3 to 7 are Johnson's approximation algorithm whose time complexity is $O(mn)$ [6]. Where *m* is the cardinality of the set that should be covered, i.e. the number of basic edge segments, and *n* is the number of subsets, i.e. the number of vertices of the given simple polygon. Thus the complexity of steps 3 to 7 is $O(n^3)$, this leads to a total time complexity of $O(n^3)$.

Algorithm-1. Vertex guarding for the edge-covering

1. Compute the visibility polygons of the vertices of the given simple polygon. Compute the basic edge segments by finding the intersections of the boundaries of the visibility polygons. Assuming $l_1, l_2, ... l_m$ are the basic edge segments, we define $E = \bigcup_{i=1}^{m} l_i$, which the set of the edges of the polygon is also.
2. For every $1 \le j \le n$, we define F_j be the set of all the basic edge segments that are completely visible from v_j. Let N= {1, 2, 3, ..., n}, and $Q = \phi$
3. Find $i \in N$ such that $|F_i| \ge |F_j|$ for all $j \in N$ and $i \ne j$.
4. Add *i* to *Q* and delete *i* from *N*.
5. For all $j \in N$, let $F_j = F_j - F_i, E = E - F_i$.
6. If $E \ne \phi$ go to 3.
7. Output the set *Q* and stop.

Let consider the edge guard version of the edge-covering problem.

Lemma 2. Every basic edge segment is completely visible from an edge of the polygon.

Proof. Let us assume on the contrary that there is a basic edge segment l that is partially visible from edge e. Thus there are two points *a* and *b* on l such that edge e sees *a* but not *b*. Thus line segment $\overline{ab}$ intersects one of the constructed edges of the weak visibility polygon of edge e. Every constructed edge of a weak visibility polygon lies on a line passes through two vertices of the polygon. This means that a line passing through two vertices of the polygon divides a basic line segment, a contradiction. □

Thus, the edge guard version of the edge-covering problem can be considered as an instance of the set covering problem. The set that should be covered is the set of the basic edge segments, and the subsets are the sets of the basic edge segments that are weakly visible from an edge of the polygon.

The proposed algorithm for vertex guard version can be simply modified for edge guard version. To this end, in the algorithm the weak visibility polygons of the edges should be used instead of the visibility polygons of the vertices. The weak visibility

polygon of an edge of a simple polygon can be computed in linear time [8]. Thus the time complexity of the modified algorithm remains unchanged, and it is also $O(n^3)$.

3 Conclusion

In this paper we investigated the edge-covering problem. In this problem the guards should patrol the edges of a given simple polygon. We considered two versions of the problem, the vertex guard and the edge guard versions. We proposed two approximation algorithms with logarithmic factor for these two versions.

References

1. O'Rourke, J.: Art Gallery Theorems and Algorithms. Oxford University Press, Oxford (1987)
2. Shermer, T.: Recent Results in Art Gallery. IEEE Proc. 80(9), 1384–1399 (1992)
3. Urrutia, J.: Art Gallery and Illumination Problems. In: Sack, J.R., Urrutia, J. (eds.) Handbook of computational geometry, Internal Report, Ottawa University, Canada (1996)
4. Laurentini, A.: Guarding the Walls of an Art Gallery. The Visual Computer 15, 265–278 (1999)
5. Cormen, T.H., Leiserson, C.E., Rivest, R.L.: Introduction to Algorithms, 2nd edn. MIT Press, Cambridge (2001)
6. Johnson, D.S.: Approximation Algorithms for Combinatorial Problems. J. Computer and System Sciences 9, 256–278 (1974)
7. EL Gindy, H., Avis, D.: A linear Algorithm for Computing the Visibility Polygon from a Point. J. Algorithms 2, 186–197 (1981)
8. Guibas, L., Hershberger, J., Leven, D., Sharir, M., Tarjan, R.E.: Linear Time Algorithms for Visibility and Shortest Path Problems inside Simple Polygons. In: 2nd ACM Symposium on Computational Geometry, pp. 1–13. ACM Press, New York (1986)

A Novel Crosstalk Estimator after Placement*

Arash Mehdizadeh[1] and Morteza Saheb Zamani[2]

[1,2] Computer Engineering Department, Amirkabir University of Technology, Tehran, Iran
{a_mehdizadeh,szamani}@aut.ac.ir

Abstract. In this paper, we propose a probabilistic method to estimate intra-grid wirelength of nets after placement or global routing. Results of incorporating this method in the previous probabilistic coupling capacitance and crosstalk estimation scheme show its efficiency in detecting noisy nets before having detailed information of wire adjacency. Our general method improved the number of correctly detected noisy nets by 15% on average. In a second improvement, the directed version of this method increased the number of correct detections by 19% and decreased the number of false detections.

Keywords: Physical Design, Placement, Routing, Coupling Capacitance, Crosstalk Noise.

1 Introduction

Nowadays, crosstalk noise introduced by coupling capacitance became one of the most important issues of design closure. Due to the giga-scale number of elements in the recent designs, crosstalk mitigation should be considered in the earlier physical design stages, especially placement. There have been a number of studies on wiring congestion reduction during placement [4], [5], [6], yet just a few deal with crosstalk explicitly [1]. In addition to the coupling capacitance, other parameters such as driver strength and coupling location affect the crosstalk noise significantly [3]. The 2-pi model from [3] is an elegant analytical model which takes into consideration many of the first-order parameters such as coupling capacitance and driver and wire resistance. To use this model, these parameters must be available which some directly depend on the precision of routing estimation.

To enhance the crosstalk estimation schemes during or after placement, where there is not enough routing information, we introduce a probabilistic method to estimate the detailed routing results inside global routing grids. Using our method, we can derive approximate length of wire-segments of each net inside every global routing grid. Feeding this information to an efficient wire adjacency and crosstalk estimation method like [2], we can extract the first order parameters of the 2-pi crosstalk model. Hence, it is possible to derive the crosstalk map and use it as guidance for crosstalk reduction.

* This work is supported by Iran Telecommunications Research Center (ITRC).

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 934–937, 2008.

2 Intra-grid Routing Estimation without Internal Connection

Consider that we have the global routing estimation or information of the nets traversing global grids. For a horizontal traversing net (Fig.1. a), the two intersections on the vertical left and right edges can be placed in different vertical distances. We call this a *horizontal 2-pass* net. If the width and length of the grid are *W* and *L* respectively, this vertical distance can vary from 0 to *L*. We call this *conditional vertical distance* (CVD). This net should also traverse the horizontal distance from one edge to the other one. This horizontal distance is equal to *W* and we call it the *forced horizontal displacement* (FHD).

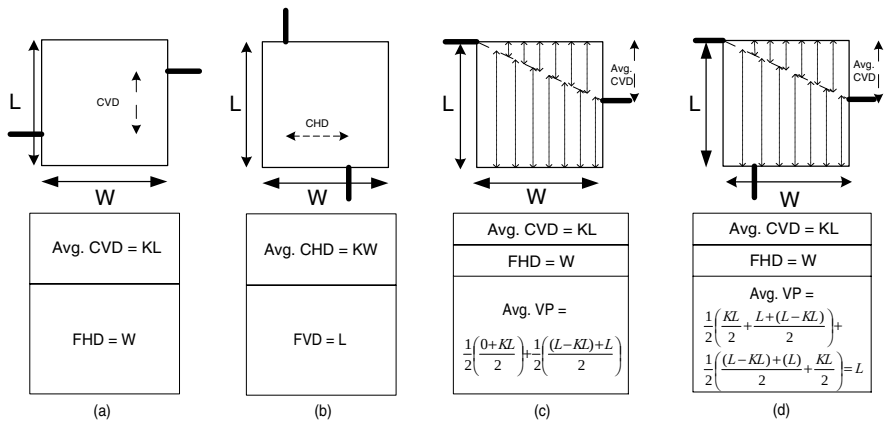


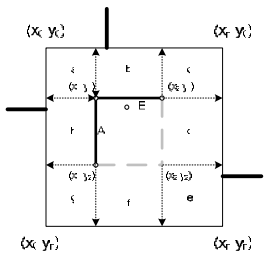
Fig. 1. Wirelength computation for (a) horizontal 2-pass net, (b) vertical 2-pass net, (c) 3-pass net, (d) 4-pass net

Another case in which a net traverses the grid vertically is shown in Fig.1. b. We call this a *vertical 2-pass* net. If the boundary pins can be placed on any of the tracks of the edges with equal chance, it can be proved that the average values of coefficients *K* and *K'* will be equal to 0.5. These values are a little pessimistic which can lead to false detections. As a result, one can examine the length of horizontal and vertical segments in these cases for a set of different designs with a specific detailed router and extract average values for the *K* and *K'* coefficients. We did this practice for a set of 6 test circuits using the router proposed in [7]. The results are presented in section 4. Having the average wirelength of horizontal and vertical segments of the 2-pass nets, we can analyze more sophisticated cases in which the traversing nets pass more than two edges of the grid (Fig.1. c, d). In these cases *vertical/horizontal paces* (VP/HP) will be added to the conditional vertical/horizontal displacements respectively.

3 Intra-grid Routing Estimation with Internal Connection

There are also cases in which there is one or more than one point in a grid that belong to one net. In Fig. 2 (x_0,y_0), (x_n,y_n), (x_1,y_1), etc. show the coordinates of different points

in the grid area. In Fig. 2, there are four internal points all belonging to the same net which has also to be connected to the external connections going through the left, right and up edges. Connection of internal points is estimated with a half-perimeter routing (here, with two segments named *A* and *B*).



$$Left(A): \left\{\left\{\frac{(y_2-y_1)}{y_n-y_0}(x_1-x_0)\Big|_X\right\}_h + \left\{\frac{y_1-y_0}{y_n-y_0}\left[\frac{y_1-y_0}{2}\Big|_Y + (x_1-x_0)\Big|_X\right]\right\}_a + \left\{\frac{y_n-y_2}{y_n-y_0}\left[\frac{y_n-y_2}{2}\Big|_Y + (x_1-x_0)\Big|_X\right]\right\}_g \right. \quad (1)$$

$$Up(B): \left\{\frac{(x_2-x_1)}{x_n-x_0}(y_1-y_0)\Big|_Y\right\}_b + \left\{\begin{array}{l}\frac{y_1-y_0}{y_n-y_0}\times\frac{x_1-x_0}{x_n-x_0}\left[\frac{3}{4}(y_1-y_0)\right]\Big|_Y \\ +\left(1-\frac{y_1-y_0}{y_n-y_0}\right)\times\frac{x_1-x_0}{x_n-x_0}\left[\frac{(x_1-x_0)}{2}\Big|_X + (y_1-y_0)\Big|_Y\right]\end{array}\right\}_a + \left\{\frac{x_n-x_2}{x_n-x_0}\left[\frac{(x_n-x_2)}{2}\Big|_X + (y_1-y_0)\Big|_Y\right]\right\}_c \quad (2)$$

$$Right(A): \frac{x_n-x_2}{y_n-y_0}\Big|_X + \left\{\begin{array}{l}\frac{y_1-y_0}{y_n-y_0}\times\frac{x_n-x_2}{x_n-x_0}\times[3/4(y_1-y_0)]\Big|_Y \\ +\left(1-\frac{x_n-x_1}{x_n-x_0}\right)\times\frac{y_1-y_0}{y_n-y_0}\left[\frac{(y_1-y_0)}{2}\Big|_Y + (x_n-x_1)\Big|_X\right]\end{array}\right\}_c + \left\{\frac{y_n-y_1}{y_n-y_0}\left[\frac{(y_n-y_1)}{2}\Big|_Y + (x_n-x_2)\Big|_X\right]\right\}_{d,e} \quad (3)$$

Fig. 2. Computing wirelength with three external and four internal connections

Equations 1-3 calculate the cost of connecting the half-perimeter to the edges. We used *X* and *Y* indices to correspond a value to a wire-segment in the horizontal or vertical direction respectively. In these equations the wire-segment nearest to a specific edge is chosen for connection of the whole connected component to that edge. For example the Equation denoted by *Left*(*A*), computes average wirelength of connecting the half-perimeter connection to the left edge (which is through segment *A*). Similar analysis can be conducted for the other patterns of traversal while having internal connections. Due to the similarity, we did not repeat them here.

4 Experimental Results and Conclusion

To appraise our noise estimation methodology, we used routing feedbacks of a multi-level routing framework introduced in [7]. Six test circuits of [7] were global and detailed routed by the specified router. Table 1 presents the characteristics of these six circuits along with the accuracy results of the following three different crosstalk estimation schemes all using the method used in [2] for basic probabilistic coupling capacitance estimation: A) Without using our proposed wirelength estimation method. B) Using our methodology and setting *K* and *K'* coefficients to the general value 0.5 (general method). C) Using our methodology while measuring average values of *K*

Table 1. Test circuits' specifications with results of different crosstalk estimation schemes

Circuit	Size (um)	# of Nets	# of Noisy Nets	Correct Detections			False Detections		
				A	B	C	A	B	C
S5378	4330*2370	3124	53	39	47	50	22	32	17
S9234	4020*2230	2774	45	30	38	38	17	20	14
S13207	6590*3640	6995	122	91	107	112	43	55	37
S15850	7040*3880	8321	140	107	124	132	54	63	35
S38417	11430*6180	21035	300	225	273	285	150	230	140

and *K'* based on the results of the detailed router (directed version). The *# of Nets* column contains the number of two-pin nets in each of the six circuits after a decomposition stage. After routing, information over the complete layout is given to the extraction tool to determine the real *# of Noisy Nets*.

Correct Detections column represents the number of correctly detected noisy nets in the aforementioned three cases. In order to obtain proper values for *K* and *K'*, we examined horizontal/vertical conditional displacement of all 2-passing nets in all the test cases after detailed routing, using global routing bins of size 30pitch*30pitch. The distribution of these two coefficients depends on such factors as grid size and congestion. Nevertheless, the derived averages (K=0.2, K'=0.17) improved the general method's Correct Detections while decreased *False Detections* (the number of falsely detected noisy nets). As it is shown, the proposed approach (columns B and C) increases correct detections by 15% on average. On the other hand, the general method (column B) has more false detections due to its pessimistic estimations of the wirelength. However, this pessimism is ameliorated in the directed method (column C) by using more realistic coefficients.

References

1. Ren, H., Pan, D.Z., Villarrubia, G.: True Crosstalk Aware Incremental Placement with Noise Map. In: IEEE/ACM International Conference on Computer Aided Design, pp. 402–409 (2004)
2. Wu, D., Hu, J., Mahapatra, R.: Coupling Aware Timing Optimization and Antenna Avoidance in Layer Assignment. In: International Symposium on Physical Design, pp. 20–27 (2005)
3. Cong, J., Pan, D.Z., Srinivas, P.V.: Improved Crosstalk Modeling for Noise Constrained Interconnect Optimization. In: Asia and South Pacific Design Automation Conference, pp. 373–378 (2001)
4. Brenner, U., Rohe, A.: An effective congestion driven placement framework. In: International Symposium on Physical Design, pp. 6–11 (2002)
5. Caldwell, E., Kahng, A.B., Markov, I.L.: Can recursive bisection alone produce routable placements? In: Design Automation Conference, pp. 477–482 (2000)
6. Yang, X., Choi, B.-K., Sarrafzadeh, M.: Routability-Driven White Space Allocation for Fixed-Die Standard-Cell Placement. IEEE Transactions on Computer Aided Design of Integrated Circuits 22, 410–419 (2003)
7. Ho, T.Y., Chang, Y., Chen, S., Lee, D.: Crosstalk- and Performance-Driven Multi-level Full-Chip Routing. IEEE Transactions on Computer Aided Design of Integrated Circuits 24, 869–878 (2005)

A New Operator for Multi-addition Calculations

Kooroush Manochehri, Saadat Pourmozaffari, and Babak Sadeghian

Department of Computer Engineering & IT, Amirkabir University of Technology, Tehran, Iran
{kmanochehri,saadat,basadegh}@ce.aut.ac.ir

Abstract. Today multi operand addition is used in many aspect of computer arithmetic such as multiplication, exponentiation, etc. One of the best method for multi addition is Carry save adder that has no carry propagation during intermediate summation. This paper introduce a new method that has a performance like carry save adder for multi-addition but has fewer gates than it. This architecture can reduce the number of logic gates by 40%.

Keywords: Multi-addition, Computer arithmetic, Bitwise subtraction, Carry save adder, Public key cryptography, Exponentiation, Multiplication, Booth recoding.

1 Introduction

Exponentiation and multiplication of large integers is the basic of several well known cryptographic algorithms such as RSA [1], Elliptic Curve cryptography (ECC) [2,3], NTRU [4,5], Etc. As a result methods which speed up implementations of multiplication and exponentiation are of considerable practical significance [6,7,8,9,10,11]. Methods like Montgomery [6] and RNS [12,13,14] and using redundant number are some examples of this try [15]. In the lower level of all of these techniques, they need multi operand addition. One of the best methods for improving multi addition calculation is Carry Save Adder (CSA) that has no carry propagation in intermediate summation [10,11,16,17,18].

Booth recoding is one of the methods to decrease the hamming weight in operands that leads to increase multiplication speed [16]. The basic idea of this paper is taken from this method but it is used for multi-addition. In proposed method by using the bitwise subtraction, the speed of the summations is improved, that its performance reaches to CSA but with fewer logic gates (60% of the CSA gates).

2 Booth Recoding

The purpose of booth recoding is reducing the number of non-zero digits of operands, to reduce the time needed to calculate the multiplication. For example radix-2 booth recoding tries to reduce the number of one and changes most of them to zero because by this work the number of partial products in multiplier is reduced [16].

An example of this recoding is as follow:

1000111111010111 operand x
-100100000-11-1100-1 Recoded version y

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 938–941, 2008.

3 Bitwise Subtraction and Its Usages

If one looks at Booth recoding technique precisely, he can find that if one subtract x_{i-1} from x_i the result will be y_i or booth-recoded version. By this fact one can define a operator named as bitwise subtraction and shown by symbol ∅ to recode a number. As known the recoded version has the value of the original number but with signed digit number, so one can say that x=2x∅x. Generally by bitwise subtraction, a number x can be shown by two numbers x1 and x2 that x=x1∅x2. So by this relation one can freely change and set the bits of x1 and x2 as he wants. Table 1 shows relation between bits of x, and x1 and x2 for selecting proper coding.

Table 1. Coding for bitwise subtraction

	Available coding	
x=x1∅x2	x1	x2
0	0	0
0	1	1
1	1	0
1	0	-1

As can be seen bitwise subtraction has following relation with addition:

$$(a∅b)+c=(a+c)∅b \tag{1}$$

By this relation one can add two numbers with more speed. For this purpose he can change one of two numbers to appropriate number that when using it for addition its speed is improved. To propose a new operator for multi addition, one should assume that one of the operands is in bitwise subtraction code. So by this assumption each digit of one of the operands may be 0, 1 or -1. For reaching to fast adder one can change the code to annihilate the carry propagation. Also to prevent the growth of the digit value, the way of coding must be changed to prevent of appearing -1 in two parts of bitwise subtraction code. By this way the final result's digit will be 0 or 1 or -1. If we name the partial result of multi-operand addition as S, that its digits are 0,1 or -1, and assume that this partial result should be added with new number y, to annihilate the carry propagation we define another operand named R and act as below:

- If S_i=-1 no carry propagation is done, because y_i=0 or 1 and by adding with S_i, the result will be -1 or 0 that these digits do not propagate carry, so R_i is set to 0.
- If S_i=0 or S_i=1, to annihilate carry propagation the corresponding bit y_i should be 0 or 1 respectively. To do so by using table 1 one can assign the proper coding by considering that R shouldn't have digit -1. For instance if y_i=1 and S_i=0, S_i should become 1, so one can do that by S_i=1 and R_i=1. Or if y_i=0 and S_i=1, instead of changing S_i, y_i can be changed to 1 and R_i=1. Note that by this act in the final result the value is not changed.

After this recoding the sum of the new addition can be found by S+y (that has no carry propagation) and then bitwise subtracted by R ((S+y)∅R).

4 Architectures for Proposed Method

Like what said in the proposed method, the operand can be broken into two parts named S and R. One can show negative digit by putting 1 in same bit position of R. Suppose that the new operand that should be summed with the result, named x, so the carry of each bit is generated when each of S_i or x_i was 1, because according to proposed method in previous section when one of operand is 1 the other become 1 and corresponding bit in R is set to 1. Remember that each bit 1 in R means -1. So one can write the Boolean relation of the new bit of R (R_i*) as follow:

$$R_i^* = x_i\overline{R_i}\,\overline{S_i} + x_i R_i S_i + \overline{x_i}\,\overline{R_i} S_i + \overline{x_i}\,\overline{R_i} S_i = x_i \oplus R_i \oplus S_i \tag{2}$$

S_i can be equal to the carry of previous position. According to proposed method if digit of each of the operands was 1, carry is generated. To show that, one can check that if x_{i-1}=1 or S_{i-1}=1 carry become 1, unless R_{i-1}=1. So the Boolean relation can be:

$$S_i^* = (x_{i-1} + S_{i-1})\overline{R_{i-1}} \tag{3}$$

The hardware block to calculate the bits of S and R is shown in Fig. 1.

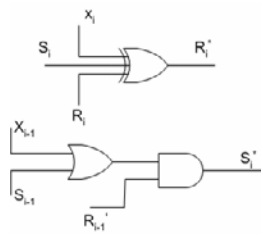


Fig. 1. Logic to find the result of each digit according to the method

To compare with Full adder block that is used in CSA architectures, this block has two AND gate less than it and if one uses this block instead of Full adder, he can save 40% of area. To compute binary result from bitwise subtraction code, BRFA like Fig. 2 can be used.

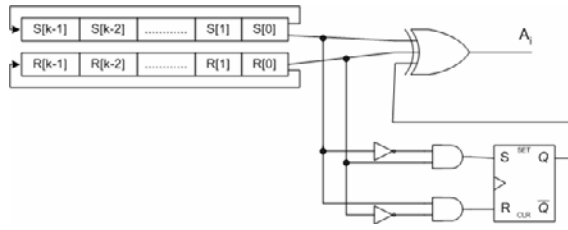


Fig. 2. Modified BRFA for proposed method

5 Conclusion

This paper proposed architecture like CSA but with fewer gates. In this method redundancy used in other view, and by bitwise subtraction, it annihilates carry propagation. This yields a method that has no carry propagation like CSA but with 40% less area. This viewpoint of redundancy can be used for multiplier and exponentiation architecture too.

References

1. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM 21, 120–126 (1978)
2. Koblitz, N.: Elliptic curve cryptosystems. Mathematics of computation 48, 203–209 (1987)
3. Miller, V.S.: Use of elliptic curve in cryptography. In: Williams, H.C. (ed.) CRYPTO 1985. LNCS, vol. 218, pp. 417–428. Springer, Heidelberg (1986)
4. Hoffstein, J., Pipher, J., Silverman, J.: NTRU: A ring-based public key cryptosystem. In: Buhler, J.P. (ed.) ANTS 1998. LNCS, vol. 1423, pp. 267–288. Springer, Heidelberg (1998)
5. Lee, M.K., Kim, J.W., Song, J.E., Park, K.: Sliding window method for NTRU. In: Katz, J., Yung, M. (eds.) ACNS 2007. LNCS, vol. 4521, pp. 432–442. Springer, Heidelberg (2007)
6. Montgomery, P.L.: Modular multiplication without trial division. Mathematics of computation 44, 519–521 (1985)
7. Bernal, Guyot, A.: Design of a modular multiplier based on Montgomery's algorithm. In: 13th conference on design of circuits and integrated systems, Madrid, Spain, pp. 680–685, November 17-20 (1998)
8. Blum, T., Paar, C.: Montgomery modular exponentiation on reconfigurable hardware. In: Proc. 14^{th} IEEE symp. on computer arithmetic, pp. 70–77 (1999)
9. Ching-Chao, Chang, T.-S., Jen, C.-W.: A new RSA cryptosystem hardware design based on montgomery's algorithm. IEEE transactions on circuits and systems 45, 908–912 (1998)
10. Manochehri, K., Pourmozafari, S.: Modified radix-2 Montgomery modular multiplication to make it faster and simpler. In: ITCC 2005, vol. 1, pp. 598–602. IEEE Computer Society, Los Alamitos (2005)
11. Manochehri, K., Pourmozafari, S.: Fast Montgomery modular multiplication by pipelined CSA architecture. In: ICM 2004, pp. 144–147. IEEE, Los Alamitos (2004)
12. Bajard, J.C., Imbert, L.: A full RNS implementation of RSA. IEEE Transaction on Computers, 769–774 (2004)
13. Pearson, D.: A parallel implementation of RSA, Cornell University (July 1996)
14. Posch, K.C., Posch, R.: Residue number system: a key to parallelism in public key cryptography. IEEE, Los Alamitos (1992)
15. Crookes, D., Jiang, M.: Using signed digit arithmetic for low-power multiplication. Electronic letters 43(11), 613–614 (2007)
16. Parhami, B.: Computer arithmetic. Oxford university press, Oxford (2000)
17. McIvor, C., McLoone, M., McCanny, J.V., Daly, A., Marnane, W.: Fast Montgomery modular multiplication and RSA cryptographic processor architectures, Department of electrical and electronic engineering, University college Cork (2003)
18. Manochehri, K., Pourmozafari, S., Sadeghian, B.: Efficient methods in converting to Modulo 2^n+1 and 2^n-1. In: ITNG 2006. IEEE Computer Society, Los Alamitos (2006)

Quantum Differential Evolution Algorithm for Variable Ordering Problem of Binary Decision Diagram

Abdesslem Layeb and Djamel-Eddine Saidouni

Lire laboratory, infography group, university Mentouri of Constantine Algeria
Layeb.univ@gmail.com, saidounid@hotmail.com

Abstract. This paper presents a new variable ordering method called QDEBDD to reduce the size of Binary Decision Diagram (BDD). The size of BDD is very reliant on the order of function variables. Unfortunately, the search for the best variables ordering has been showed NP-difficult. In this work, the variable ordering problem is cast as an optimization problem for which a new framework relying on quantum computing is proposed. The contribution consists in defining an appropriate quantum representation scheme that allows applying successfully on BDD problem some quantum computing principles. This representation scheme is embedded within a Differential Evolution Algorithm leading to an efficient hybrid framework which achieves better balance between exploration and exploitation capabilities of the search process.

Keywords: Quantum Computing, BDD, Differential Evolution Algorithm.

1 Introduction

The Binary Decision Diagrams BDD is rooted directed acyclic graph. The BDD yields a data structure for Boolean functions that has been proven to be very useful in many fields of computer. However, the major problem with BDD size is the function variables order. Therefore, it is important to find a variable order which minimizes the number of nodes in a given BDD. Unfortunately, this task is not simple taking into account the fact that there is an exponential number of possible variable ordering. Indeed, this problem was showed NP-difficult [1].

Quantum computing is a new theory which has emerged as a result of merging computer science and quantum mechanics. A particle in the quantum mechanics can be in a superposition of states. By taking account of this idea, one can define a quantum bit or the qubit which can take value 0, 1 or a superposition of the two at the same time. Its state can be given by:

$$\Psi = \alpha\,|0\rangle + b|1\rangle \tag{1}$$

Where $|0\rangle$ and $|1\rangle$ represent the classical bit values 0 and 1 respectively; a and b are complex numbers such that.

$$|\alpha|^2 + |b|^2 = 1 \tag{2}$$

The probability that the qubit collapses towards 1 (0) is $|\alpha|^2$ ($|b|^2$).This idea of superposition makes it possible to represent an exponential set of states with a small

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 942–945, 2008.

number of qubits. According to the quantum laws like interference, the linearity of quantum operations and entanglement make the quantum computing more powerful than the classical machines [2].

Differential Evolution (DE) is new algorithm of optimization, it's inspired by the genetic algorithms and the evolutionary strategies combined with a geometrical technique of research. It was introduced by Storn and Price in 1996 [3]. Each solution is encoded with n-dimensional vector based on floating-point numbers $\vec{a} \in R^n$ and the whole population of potential solutions at generation g, is given by $P(g) = \{\vec{a}_i, ..., \vec{a}_m\}$. The size of the Population *m* does not change during the minimization process. At each generation, new vectors are generated by the combination of vectors randomly chosen from the current population. The novel in DE algorithm is the mutation operator. In DE, the mutation operator mutates *m* vectors through the weighted difference of two (or four) other vectors according to the following formula [4]:

$$\vec{v}_i = \vec{a}_{r1} + F \times (\vec{a}_{r2} - \vec{a}_{r3}) \tag{3}$$

where i =1,2,..., m , and the random indexes r1, r2, r3 ∈ [1,2,..., *m*] are mutually different and also distinct from the index *i*. $F \in [0,2]$ is a real constant which affects the differential variation between two vectors.

2 The Proposed Approach

The problem consists to define the best permutation of the Boolean function variables V={X_1, X_2… X_n), that gives the minimal BDD size. In order to easily apply quantum principles on variable ordering problem, we need to map potential solutions into a quantum representation that could be easily manipulated by quantum operators. The variable order is represented as binary matrix. In term of quantum computing, all feasible variable orders can be represented by a quantum matrix *QM* (fig.1) that contains the superposition of all possible variable permutations. Each pair (α_i, b_i) represents a single qubit and corresponds to the binary digit 1 or 0. For each qubit, a binary value is computed according to its probabilities $|a_i|^2$ and $|b_i|^2$.

Given a set *V* of BDD variables, an initial population *PQM* of solutions is encoded in N quantum chromosomes. The algorithm QDEBDD consists on applying iteratively the quantum operators to generate other solutions. The first operation is the interference operation (fig.2). This one amplifies the amplitude of the best solution and decreases the amplitudes of the bad ones. It primarily consists in moving the state of each qubit in the direction of the corresponding bit value in the best solution in progress [5]. This operation can be accomplished by using a unit transformation which achieves a rotation whose angle is a function of the amplitudes a_i, b_i and of the value of the corresponding bit in the solution reference. Next, we apply the Quantum Differential Mutation. It consists in perturbing values of quantum chromosomes using a difference between two other quantum chromosomes. In the first step of this mutation we make the difference between two chosen chromosomes. Then we add this difference to the third chromosome. Finally, we perform a selection of the chromosomes having the best fitness value. For this, we apply first a measurement on each

chromosome to have from it one solution among all those present in superposition. This operation transforms by projection the quantum matrix into a binary matrix (fig.3). Therefore, there will be a solution among all the solutions present in the superposition. The evaluation of the solutions is done by using the size of the BDD obtained from the order as criterion of selection. Global best solution is then updated if a better one is found and the whole process is repeated until having satisfaction of a stopping criterion.

$$\begin{bmatrix} \begin{pmatrix} a_{11} \\ b_{11} \end{pmatrix} & \begin{pmatrix} a_{12} \\ b_{12} \end{pmatrix} & \cdots & \begin{pmatrix} a_{1m} \\ b_{1m} \end{pmatrix} \\ & \vdots & & \\ \begin{pmatrix} a_{n1} \\ b_{n1} \end{pmatrix} & \begin{pmatrix} a_{n2} \\ b_{n2} \end{pmatrix} & \cdots & \begin{pmatrix} a_{nm} \\ b_{nm} \end{pmatrix} \end{bmatrix}$$

Fig. 1. Quantum representation of variable ordering

$$\begin{pmatrix} \begin{pmatrix} 0.99 \\ 0.14 \end{pmatrix} & \begin{pmatrix} 0.3778 \\ -0.9259 \end{pmatrix} & \begin{pmatrix} 0.14 \\ 0.99 \end{pmatrix} & \begin{pmatrix} 0.8884 \\ -0.459 \end{pmatrix} \\ \begin{pmatrix} 0.14 \\ 0.99 \end{pmatrix} & \begin{pmatrix} 0.3778 \\ -0.9259 \end{pmatrix} & \begin{pmatrix} 0.99 \\ 0.14 \end{pmatrix} & \begin{pmatrix} 0.14 \\ 0.99 \end{pmatrix} \\ \begin{pmatrix} 0.14 \\ 0.99 \end{pmatrix} & \begin{pmatrix} 0.14 \\ 0.99 \end{pmatrix} & \begin{pmatrix} 0.3778 \\ -0.9259 \end{pmatrix} & \begin{pmatrix} 0.8884 \\ -0.459 \end{pmatrix} \\ \begin{pmatrix} 0.8884 \\ -0.459 \end{pmatrix} & \begin{pmatrix} 0.8884 \\ -0.459 \end{pmatrix} & \begin{pmatrix} 0.14 \\ 0.99 \end{pmatrix} & \begin{pmatrix} 0.99 \\ 0.14 \end{pmatrix} \end{pmatrix} \xrightarrow{\textbf{Measure}} \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Fig. 2. Quantum measurement

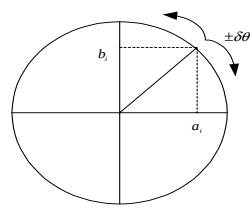


Fig. 3. Quantum interference

3 Implementation and Evaluation

QDEBDD is implemented in java 1.5 and is tested on a microcomputer with a processor of 2 GHZ and 256 MO of memory. For the creation of BDD and the evaluation of the fitness of each BDD, We have used the package JBDD [6]. To assess the efficiency and accuracy of our approach several experiments were designed. In all experiments, the size of population is 4, the permutation probability is a tunable

parameter which was set to 0.15, the interference angle is $\pi/20$, and the iteration numbers vary between 100 iterations for small tests and 1000 iterations for large tests. The found results (table1) are promising and prove the feasibility and the efficiency of our approach.

Table 1. Results given by QDEBDD

Test	Number of variables	Initial Solution	Final Solution
Test1	200	551	301
Test2	300	861	497
Test3	300	716	442
Test4	350	915	527
Test5	400	1054	605
Test6	200	324	200
Test7	250	402	250
Test8	300	487	300
Test9	350	582	351
Test10	400	659	400

4 Conclusion

In this paper, we have presented a new approach named QDEBDD to deal with the problem of variable ordering in the BDD. QDEBDD is based on a hybridizing of quantum computing principles and differential evolution algorithm. The quantum representation of the solutions allows the coding of all the potential variable orders with a certain probability. The experimental studies prove the feasibility and the effectiveness of our approach. The proposed algorithm reduces efficiently the population size and the number of iterations to have the optimal solution.

References

[1] Bollig, B., Wegener, I.: Improving the variable ordering of OBDDs is NPcomplete. IEEE Trans. on Comp. 45(9), 993–1002 (1996)
[2] Williams, C.P., Clearwater, S.H.: Explorations in quantum computing. Springer, Berlin (1998)
[3] Storn, R., Price, K.: Differential Evolution-a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11, 341–359 (1997)
[4] Draa, A., Batouche, M., Talbi, H.: A Quantum-Inspired Differential Evolution Algorithm for Rigid Image Registration. In: The proc. of ICCI, Turkey, pp. 408–411 (2004)
[5] Layeb, A., Meshoul, S., Batouhe, M.: Multiple Sequence Alignment by Quantum Genetic Algorithm. In: The proc. of the 20th IPDPS, Greece, pp. 1–8 (2006)
[6] Whaley, J.: JAVABDD, a Java Binary Decision Diagram Library, Stanford University, `http://javabdd.sourceforge.net/`

A Joint Source-Channel Rate-Distortion Optimization Algorithm for H.264 Codec in Wireless Networks

Razieh Rasouli[1], Hamid R. Rabiee[1], and Mohammad Ghanbari[1,2]

[1] AICTC Research Center, Computer Engineering Department, Sharif University of Technology, Tehran, Iran
rasouli@ce.sharif.edu, rabiee@sharif.edu
[2] Departments of Electronic Systems Engineering, University of Essex, Colchester, UK
ghan@essex.ac.uk

Abstract. In recent years, the demand for video transmission over wireless communication networks is growing very fast. The H.264 video compression standard which offers high quality at low bit rates, is a suitable codec for applications that require efficient video transmission over wireless networks. While the compressed videos are transmitted through error-prone networks, error robustness becomes an important issue. In this paper, a joint source-channel Lagrange optimization method in which the distortion of the decoder is estimated without using feedback which can be used for both multicast and point-to-point applications is proposed. The experimental results show that the new algorithm has a good performance in video transmission over error-prone channels by concealing the lost packets at the decoder.

Keywords: Wireless Networks, H.264, Lagrange Rate-Distortion Optimization, Source-Channel coding.

1 Introduction

The demand for fast and location-independent access to multimedia services offered on today's wireless networks is steadily increasing. Based on the business models in emerging wireless systems in which the end-user's costs are proportional to the transmitted data volume and also due to limited resources bandwidth and transmission power, compression efficiency is the main target for wireless video and multimedia applications. This makes H.264/AVC coding an attractive candidate for all wireless applications [1].

In [2] a mixture Lagrange optimization algorithm that offers different formulas for doing optimization with different macroblock coding options has been proposed. A more general algorithm is offered in [3]. It considers the mode correlation of neighboring macroblocks and besides, uses more coding parameters for optimization. The proposed algorithm in [4] can be considered as a special case of the algorithm presented in [3]. In this method, the quantization parameter and Lagrange multiplier of the mode selection are fixed and then Lagrange multiplier of motion prediction is calculated by using the algorithm of [2], and finally the best mode which minimizes the error is searched for. An iterative optimization algorithm is proposed in [5]. This algorithm performs the optimization with different values of quantization parameters,

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 946–950, 2008.

Lagrange multipliers and coding modes. The convergence of this iterative algorithm is somewhat slow. In the algorithms of [2-5] and many others, the quantization parameter of a macroblock is derived from the quantization parameter of the macroblock of the same position in the previous frame or the neighboring macroblocks. However, in [6], the proposed algorithm uses a premier motion prediction by using the quantization parameter of the previous frame and the original quantization parameter is selected based on the motion information of the macroblock. Finally, the best Lagrange mode for the least distortion is being chosen.

The algorithm which is used for bit-rate control and video streaming on wireless networks should be able to compensate both channel and source errors. In [1], after investigating the limitations of video transmission over the wireless networks a Lagrange optimization method have been proposed. In this method, besides the distortion of channel loss and decoder concealment, the distortion of coding each macroblock in the source is being considered. It utilizes the concepts of feedback in networks [7] and optimized multiple frame coding [8] to achieve rate-distortion optimization.

2 The Proposed Method

The simplicity and good performance are the main reasons that have made Lagrange optimization method suitable for H.264 reference model:

$$Min\ D\ subject\ to\ R < Rc \tag{1}$$

Where Rc is the desired target rate. The optimization of the above problem can be solved easily with Lagrange multiplier as:

$$Min\ J = D + \lambda R \tag{2}$$

Where parameter λ, is called the Lagrange multiplier. Each solution of equation (2) with a special value of λ is an optimized solution for equation (1) for a given Rc [2].

It seems that the proposed method of [8] in which the channel behavior and decoder distortions are estimated for calculating D, would be a suitable technique for wireless and real time applications. In this method the optimization equations are as follows [8]:

$$o^*_{n,m} = \arg\min_{o \in O}(\hat{D}_{n,m} + \lambda_0 p_c R_{n,m}(o)) \tag{3}$$

And:

$$\hat{D}_{n,m}(o) = P_c\, D_{n,m}{}^{(ef)}(o) + (1 - p_c) D_{n,m}{}^{(ep)}(o) \tag{4}$$

Where, O is the set of all modes, λ_0 is the Lagrange multiplier of error free coding, and distortion consists of the distortion of coding in error free case and the estimated distortion of error propagation which depends on the concealment method and channel behavior, shown by using Pc. For computing D(ep) , K samples of decoder distortion and channel behavior random variables, that are received trough feedback channel, have been considered and the D(ep) is calculated as the expected value of them.However, the algorithm introduced in [8] can not be used in multicast applications. Moreover, using feedback to determine the conditions of all the sent packets can cause more delay.

Inspired from the algorithm introduced in [8] (equations 3 and 4), we propose to estimate the distortion caused by the error propagation for each macroblock, without using a feedback channel (feedback is only used for understanding of channel statistics such as packet loss probability). We use previous slices' distortions to calculate average distortion of each kind of macroblock. We assume decoder uses a copy of the same position macroblock of the reference frame for concealing the erroneous macroblocks. By using the sum of distortion of each type of macroblock that has been saved in encoder ($D_{total,I}$, $D_{total,p}$,$D_{total,B}$), we calculate average distortion of each kind of macroblock (*I, B* and *P*) in error free case:

$$\begin{aligned} D_{ef,I} &= avg(D_{total,I}) \\ D_{ef,B} &= avg(D_{total,B}) \\ D_{ef,P} &= avg(D_{total,P}) \end{aligned} \tag{5}$$

Then we use the following objective function for optimization:

$$J = P_c \cdot D_{(ef)} + (1 - P_c) \cdot D_{(ep)} + \lambda \cdot R \tag{6}$$

In which $D_{(ef)}$ is the distortion caused by coding the current macroblock when the reference is received correctly, $D_{(ep)}$ is the error propagation distortion when the reference macroblock is missing and P_c is the probability of correctly receiving the reference which depends on the packet loss probability and is equal to 1-p.

The distortion caused by error propagation at decoder side (D_{ep}), is calculated regarding coding type of the current macroblock and its references as follows:

- If the slice type is I, since all its macroblock are from type I, $D_{(ep)}$ is calculated by considering number of neighbors (N) who participate in macroblock estimation:

$$D_{(ep)} = N \times D_{ef,I} \tag{7}$$

- If the slice type is P or B, and macroblock type is B or P, distortion of three references regarding their type will be considered:

$$D_{(ep)} = D_{ref\ 1}^{(ef)} + D_{ref\ 2}^{(ef)} + D_{ref\ 3}^{(ef)} \tag{8}$$

- If the macroblock type is I then according to the type of participant macroblocks in estimation we have:

$$D_{(ep)} = N_I \times D_{ef,I} + N_P \times D_{ef,P} + N_B \times D_{ef,B} \tag{9}$$

It is clear that using more intra macroblocks will increase the bit rate but we can reduce the bit overhead by adjusting QP to be able to have a constant bit rate in a short window. This is one of the subjects in our future work.

3 Simulation Results

We used the X.264 open source codec (version 0.47.534) [11] to implement our algorithm because of its speed, which is about 50 times more than JM (version 10.2), and its bit rate which is 5% lower than JM for the same PSNR [12]. We have used these settings for coding: first QP is set to 25, number of references is 5, the CPU optimizations were turned off, GOP format is IPPP.., each NAL contains one slice and packet loss probability is set to 0.2 for CIF sequences and 0.1 for QCIF ones. For lossy

channel, we have implemented an Elliot-Gilbert channel [13] and performed each simulation (transmission and decoding) of different packet loss probability for 100 times. As decoder, we have used ffmpeg open source (version 2006-11-26).

First, we used frame coding structure in which each frame contains only one slice. The result for Silent test sequence is shown in Table 1.

Table 1. Decoded average PSNR of Silent (QCIF,100 frames) for various packet losses

Method	Packet loss probability			
	0.0001	**0.001**	**0.005**	**0.01**
New algorithm	38.78	37.04	31.93	29.52
Source coding	39.08	36.62	31.82	28.6

As it's clear in the above table, our algorithm performance is better when the error rate increases and in the best case for Silent sequence we could obtain 0.92 dB increase in PSNR. Our gain for Carphone, News, and Foreman sequences were up to 1.82 dB, 0.34 dB and 0.62 dB, respectively.

For comparing our method and conventional source coding with the same condition, we tuned our channel in a way that the same frames of Forman sequence are lost.

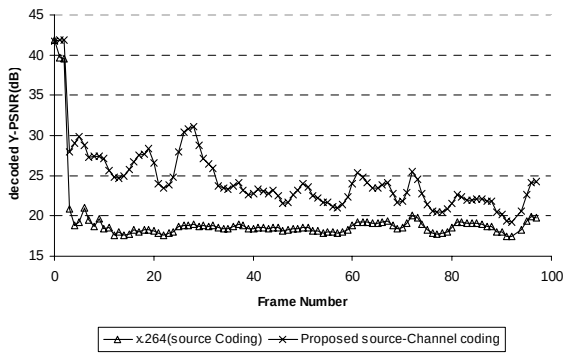


Fig. 1. Decoded PSNR of each frame when frames #3 and 34 have been lost (Foreman, CIF)

As shown in Figure 1, the PSNR difference of the two methods is about 12 dB for some frames and in average it's about 5dB which is a very good enhancement with only 11% increase in bit rate.

By adjusting QP we decreased the bit overhead caused by our algorithm to compare the two methods in the same bit rates. The results showed that our algorithm still outperforms the conventional source coding and in the best case the difference between PSNRs was about 4 dB.

For checking the effect of our method on slice structure, we increased number of slices of CIF frames to 10 and to 4 for QCIF frames. The results showed that the effect of our method on slice coding is more than its effect on frame coding and it has increased Foreman's PSNR up to 2.38 dB.

4 Conclusions

In this paper we have presented a new rate control optimization method for wireless noisy channels by using Joint Source-Channel optimization. Using Lagrange Rate Distortion (R-D) optimization, we included a coarse estimation of decoder distortion to encode video clips in a way that is more resistant against channel errors and hence achieved better quality at the decoder. Experimental results showed that in the best case, we could improve the average PSNR of decoded videos for about 1.82 dB in the presence of channel errors.

In contrast with previous methods, our algorithm does not need any feedback information. In addition, our algorithm can be used in multicast applications. Moreover, the codec complexity and delay can be reduced because there is no need to have an exact estimation of the decoder distortion or wait to receive the feedback from the decoder side.

References

1. Stockhammer, T., Hannuksela, M.M., Wiegand, T.: H.264/AVC in Wireless Environments. IEEE Transaction on CVST 13, 657–673 (2003)
2. Wiegand, T., Sullivan, G.J.: Rate-Distortion Optimization for Video Compression. IEEE Signal Processing Magazine (November 1998)
3. Wiegand, T., et al.: Efficient Mode Selection for Block-Based Motion Compensated Video Coding. In: IEEE ICIP 1995, Santa Barbara, CA (October 1995)
4. Wiegand, T., et al.: Rate-constrained Coder Control and Comparison of Video Coding Standards. IEEE Transaction on CVST 13, 688–703 (2003)
5. Ghandi, M., Ghanbari, M.: A Lagrangian Optimized Rate Control Algorithm for the H.264/AVC Encoder. In: ICIP 2004, Singapore (October 2004)
6. Pan, F., Lin, Z.: Content adaptive Rate Control. In: ICIC International, Singapore (December 2005)
7. Yu, H.B., Yu, S., Wang, C.: A highly Efficient, Low Delay Architecture for Transporting H.264 Video Over Wireless channel. Image Communications (January 2004)
8. Stockhammer, T., Kontopodis, D., Wiegand, T.: Rate-Distortion Optimization for H.26L Video Coding in Packet Loss Environment. In: Proc. Packet Video Workshop, Pittsburgh, PY (April 2002)
9. Stockhammer, T., Jenkac, H., Kuhn, G.: Streaming Video over Variable Bit-rate Wireless Channels. IEEE Transaction on Multimedia (April 2004)
10. Farber, N., Steinbach, E., Girod, B.: Robust H.263 Compatible Video Transmission Over Wireless Channels. In: Proceedings of the Picture coding Symposium, pp. 575–578 (1996)
11. X.264, `http://developers.videolan.org/x264.html`
12. Merrit, L.: X.264: A high performance H.264 Encoder. In: University of Washington, Data Compression Laboratory (2003) (unpublished)
13. Ghanbari, M.: Standard Codecs: image compression to advanced video coding. In: IEEE Telecommunications Series, London, UK (2003)

Pre-synthesis Optimization for Asynchronous Circuits Using Compiler Techniques

Sharareh ZamanZadeh, Mehrdad Najibi, and Hossein Pedram

Computer Engineering and Information Technology, Amirkabir University of Technology
Tehran, Iran
{Zamanzade,Najibi,Pedram}@aut.ac.ir

Abstract. The effectiveness of traditional compiler techniques employed in high-level synthesis of synchronous circuits aiming to present a generic code is studied for asynchronous synthesis by considering the special features of these circuits. The compiler methods can be used innovatively to improve the synthesis results in both power consumption and area. The compiler methods like speculation, loop invariant code motion and condition expansion are applicable in decreasing mass of handshaking circuits and intermediate modules. Moreover, they eliminate conditional access to variables and ports and reducing the amount of completion detection circuits. The approach is superimposed on to Persia synthesis toolset as a presynthesis source-to-source transformation phase, and results shows on average 22% improvement in terms of area and 24 % in power consumption for asynchronous benchmarks.

Keywords: Compiler techniques, Optimization, High level synthesis of asynchronous circuits.

1 Introduction

Asynchronous, modelled as message-passing networks, performs computations based on sending and receiving messages. Without global clock, synchronization is performed using local handshaking mechanisms among different communicating components. This fact potentially brings many merits for asynchronous circuits; low power consumption, average case performance, design modularity, better handling of long interconnects, avoidance of clock-related problems, robustness towards supply, process variation, and less Electro-Magnetic Interference (EMI). On the other hand, there are some drawbacks for asynchronous circuits, such as area overhead of extra handshaking circuitry and wiring to simplify timing assumptions and encoding data validity within the data itself. Area optimization is more critical in asynchronous circuits due to their higher area overhead.

We have shown that by analysing the sequence and position of instructions and manipulating them efficiently in the high-level description, a worthy optimization in area overhead and control circuit can be achieved, especially in conditional or loop bodies.

In addition both synchronous and asynchronous synthesis may suffer from dependence of the synthesis results to designers' programming styles; our method can also eliminate such dependences to produce the generic ideal specification.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 951–954, 2008.

In [1] the same problem is investigated for synchronous circuits; compiler techniques can reduce the critical path, total delay and area respectively up to 18%, 58% and 30%. In comparison we have considered our results in both area and power consumption. On average the results from our takes on are 22% improvement in area and 24% in power consumption[1] [14].

2 Asynchronous Synthesis in Persia

In the Persia tool set, asynchronous circuits are produced based on Quasi-Delay Insensitive (QDI) delay model, four phase handshaking and they are decomposed into fine grain processes which are adaptable to Pre-Charge Full/Half Buffers (PCFB/PCHB) templates [7]. The input to the synthesis tool is an asynchronous circuit description in Verilog-CSP [10] which uses READ and WRITE macros to simplify specifying asynchronous communication over the channels [11]. The Persia synthesis tool converts the high-level description to a net-list of standard cells in a step by step fashion as shown in Fig. 1.

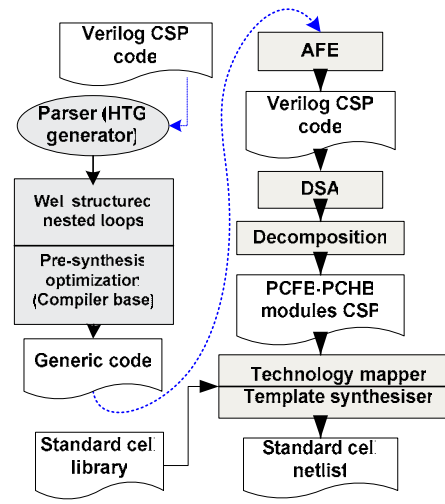


Fig. 1. Asynchronous synthesis flow

3 Complier Based Optimization for Asynchronous Circuits

Complier techniques are employed as presynthesis step prior decomposition [12]. To describe the rules of code conversions we need an intermediate representation of control and data dependencies, the pattern we have utilized is an offspring of HTG model [2] [3]. The HTG (Hierarchical Task Graph) is a directed acyclic graph with unique start and stop nodes. There is at least a path from the start node to every node in the HTG and a path from every node in the HTG to the stop node. Edges represent control flow. The main effective compiler methods that we are going to discuss below respectively are speculation, loop invariant code motion and condition expansion.

Speculation refers to the unconditional execution of operations that were originally supposed to be conditional. This method is applicable in three aspects: 1. it effects on the dominance relationship [4] among instructions. As a result it can be used in

[1] In [14] you can see that power consumption is linearly related to transition number.

converting partial redundancy to total redundancy and removing unnecessary computations. 2. Speculation can emit instructions that are both in if-clause and else-clause, we name it factorization. But there are some situations that two instructions in if/else clauses are lexically different but semantically the same, so by attention to this issue they can be considered to be factorized and be executed unconditionally. 3. There may exist some instructions that are locally anticipated at the entry point of their basic blocks and transparent in other points of the basic block , so without any side effect to the final result they can be moved to out of conditional body. [13]

Frequently, there exist computations within a loop that produce the same results each time the loop is executed. The computations are known as loop-invariant codes and can be moved outside of the loop body without changing the results of the code, they may lead to reduce power consumption or area in some cases. These codes should be executed only once instead of being executed in any loop-iteration. Since in the data driven decomposition method any loop will be smote in to two parts [12]: one part for the value of last iteration and one for the value of current iteration, by loop invariant code motion we can reduce the considerable handshaking and port numbers among these two parts in synthesized asynchronous circuits.

Condition expansion method has phenomenal effects on area and power consumption reduction. The condition of the if-statement can be conjunctively added to the assignments exist in the if-clause and inversely to those in the else clause. Using this method is possible providing there are no READ and/or WRITE operations in neither if nor else clauses. In this case conditional body can be completely substituted with unconditional blocks. This method is very useful after data driven decomposition since modules associated with conditional variables and necessary handshaking of reading and writing of *conditionally assigned variable*[2] will be removed. In nested conditional bodies, we first start from the innermost condition.

4 Conclusion and Results

The affect of the compiler code motions on various asynchronous benchmarks are considered during various stages of synthesis. The similar feature among all the discussed compiler techniques is that they lessen intermediate communications and completion detection circuits that have lead up to 60% improvement in area and 56% in power consumption. These results proof that area and power saving are highly related to reduction in communications and condition modules. At last we presented a general presynthesis algorithm to apply compiler techniques efficiently.

We selected a Reed-Solomon decoder as a test-bench. The circuit can correct two symbol (8-bit) errors. [8] [9]

The final results have been shown in table 1, with the criteria of transition and transistor amounts. The gain is on average 24% in transition numbers and 22% in the number of transistors.

[2] Those variables that are left or right hand of an assignment located in a conditional body.

Table 1. Result of Read-Solomon-and its modules

Test benches	Original Transistor#	Original Transition#	Improved Transistor#	Improved Transition#
Chienforny	38668	413	15841 (60%)	180 (56 %)
Fsyndrome	22134	238	18659 (16 %)	171 (28 %)
RiBm	81513	619	76706 (6 %)	580 (6 %)
Fifio	729	6	729 (0 %)	6 (0 %)
Reed Solomon	143044	1252	111935 (22%)	948 (24 %)

References

1. Gupta, S., Gupta, R., Dutt, N., Nicolau, A.: Coordinated parallelizing compiler optimizations and high level synthesis. ACM transitions on design of electronic systems 9(4) (October 2004)
2. Gupta, S., Savoiu, N., Dutt, N.D., Gupta, R.K., Nicolau, A.: Using global code motions to improve the quality of results for high level synthesis. IEEE. Transitions on computer aided design of integrated circuits and systems 23(2) (February 2004)
3. Nicolau, A., Novack, S.: Trailblazing: A hierarchical approach to percolation scheduling. In: International conference on parallel processing (1993)
4. Streedhar, V.C., Gao, G.R., Lee, Y.-F.: Incremental computation of dominator trees. ACM Trans. Program. Languages and syst. 19, 2 March (1997)
5. Gupta, S., Savoiu, N., Kim, S., Dutt, N., Gupta, R., Nicolau, A.: Speculation techniques for High level synthesis of control Intensive designs. In: Proceedings of the Design Automation Conference, pp. 269–272. ACM, New York (2001)
6. Tugsinavisut, A.: Design and synthesis of concurrent asynchronous systems. The dissertation for the degree doctor of philosophy (electrical engineering) in university of southern California (December 2005)
7. Asynchronous group of Amir Kabir University: Design and Implementation of Synthesis toolset for asynchronous and GALS circuits. Persia. Technical report to industrial and mines ministry of Iran
8. Saleh, K.: Hardware Architectures for Reed-Solomon Decoders. Technical Report, Amir kabir University of Technology (January 2003)
9. Wicker, S., Bhargava, K.: Reed-Solomon Codes and Their Applications. IEEE Press, Los Alamitos (1994)
10. Seifhashemi, A., Pedram, H.: Verilog HDL, a Replacement for CSP. In: 3rd ACiD-WG Worksop FP5, FORTH, Heraklion, Crete, Greece (January 2003)
11. Seifhashemi, A., Pedram, H.: Verilog HDL, Powered by PLI: a suitable Framework for Describing and Modeling Asynchronous Circuits at All Levels of Abstraction. In: Proc. Of 40th DAC, Anneheim, CA, USA (June 2003)
12. Wong, G.: High-Level: Synthesis and Rapid Prototyping of Asynchronous VLSI Systems, PhD thesis, California Institute of Technology (May 2004)
13. Morel, E., Renvoise, C.: Global optimization by suppression of partial redundancies. Communication of ACM 22(2) (February 1979)
14. Niknahad, Ghavami, Najibi, Pedram: A power estimation methodology for QDI asynchronous circuits based on high-level simulation. In: IEEE computer society annual symposium on VLSI (ISVLSI 2007), pp. 471–472 (2007)

Absolute Priority for a Vehicle in VANET

Rostam Shirani, Faramarz Hendessi, Mohammad Ali Montazeri, and Mohammad Sheikh Zefreh

Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan, 84156-83111, Iran
{rostam_sh,sheikhzefreh}@ec.iut.ac.ir,
{hendessi,montazer}@cc.iut.ac.ir

Abstract. In today's world, traffic jams waste hundreds of hours of our life. This causes many researchers try to resolve the problem with the idea of Intelligent Transportation System. For some applications like a travelling ambulance, it is important to reduce delay even for a second. In this paper, we propose a completely infrastructure-less approach for finding shortest path and controlling traffic light to provide absolute priority for an emergency vehicle. We use the idea of vehicular ad-hoc networking to reduce the imposed travelling time. Then, we simulate our proposed protocol and compare it with a centrally controlled traffic light system.

Keywords: Intelligent Transportation System, Traffic Light Control, Shortest Path Algorithm, Vehicular Ad-Hoc Network.

1 Introduction

Two important usages of Intelligent Transportation System (ITS) are shortest path (SP) finding and traffic light control (TLC). In TLC, the idea is to control the traffic lights in order to optimize a desired parameter. In this paper, our goal is to minimize the travel time of a distinguished vehicle from a source to a destination. The advantage of this protocol is when we want to provide an Absolute Priority for a vehicle. In absolute priority, all of the available network capabilities are used to impose the lowest possible travel time to an emergency vehicle such as an ambulance.

For the rest of the paper, we use ambulance as a symbolic name for each vehicle that needs absolute priority in an urban environment. Dijkstra [1], the most common SP algorithm in VANET scenario, is used to find SP. In order to have TLC, in [2], the idea is to use Geographical Information System (GIS). Huang proposed a design strategy based on statechart for TLC [3]. Furthermore other researchers have used the idea of Intelligent TLC [4]. In this paper, we propose a decentralized algorithm with simultaneous use of SP and TLC based on packet structure of Dynamic Source Routing (DSR) [5]. Our proposed structure, which acts in an emergency situation, could be used as a complement for a central traffic control system.

Our proposed protocol consists of two algorithms. One of these algorithms tries to estimate the cost of different paths towards a destination without relying on GIS or any other central data base. After that, it is necessary to have an efficient forwarding mechanism for addressing MAC issues. The aim of this mechanism is to send packets

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 955–959, 2008.

with the lowest possible delay. And finally, the most important algorithm is the one which keeps the traffic light green to prepare absolute priority for the ambulance.

2 Protocol Description

We consider that each vehicle has an ID, and ambulance ID is recognizable for traffic lights. We assume that vehicles are equipped with GPS and Digital Road Map. Therefore, it is not very hard to calculate the physical distance. But decentralized calculation of vehicles' density is more challenging. However, our method is able to intelligently recognize SP (the path with the least imposed travelling time). The packet structure is similar to a usual DSR packet which includes three additional parts: A critical tag used by an ambulance to show emergency situation, a tag which is used as a sign for other vehicles to ask them whether or not insert their characteristics, and a table including vehicle's velocity, vehicle's and street's ID, and vehicle's position based on GPS information.

In the first step, the algorithm should find the SP. The ambulance starts sending requests towards destination. Because transmission range is limited, a multi-hop mechanism should be applied. All the intermediate nodes insert their velocity, their ID, and their current street's ID in the receiving packet and then forwards the packet. When the packet reaches to the destination, it should travel back towards source. In this returning trip, intermediate vehicles act as relays. When the packets of different paths returned back to source, the cost values will be estimated.

In our design, cost is defined as travelling delay (the time that is needed for a vehicle to travel from a source to a destination); this time is different for different paths. The more the delay of a path is, the more the cost of that path will be. For estimating travel time, two parameters are needed: physical distance from source to destination, and average speed in the path. The physical distance is calculated by GPS.

In order to estimate the speed, we use an averaging mechanism. We assume that each vehicle inserts its current speed in the packet. Hence, by averaging over speeds of vehicles which are in a specific street, the cost of that street is estimated. Assume that C_i is the cost value of the street S_i. Also, L_i is the length of the street S_i. In each street several vehicles cooperate to form a multi-hop scenario, $V_i = \frac{1}{K}\sum_{k=1}^{K} V_i^k$ is the average velocity of vehicles which cooperates in message forwarding in street S_i. K is number of vehicles in S_i which participates in data forwarding. When the packet comes back to source node (ambulance), it is easy to calculate delay of different paths. Delay of the m^{th} path is shown in (1).

$$S_m \Rightarrow \begin{cases} L_m & Distance\ from\ digital\ road\ map \\ V_m & Velocity\ from\ averaging \end{cases} \Rightarrow T_m = \frac{L_m}{V_m} \quad (1)$$

As it was mentioned before, cost value of each path is defined as related delay of that path. Therefore $T_1, T_2 \ldots T_N$ could be used as cost values. In other words, $C_i = T_i$ is the cost value of street S_i. A summation over the streets' delay of a path gives the cost of that path. Therefore, Dijkstra algorithm uses these cost values for finding SP. In the following algorithm, the task of source is shown.

Step 1- setting the critical bit
Step 2- setting velocity tag
Step 3- sending a request towards destination coordination (intermediate vehicles insert their velocity in the corresponding part of the packet towards destination)
Step 4 -waiting for arrival of replay packet
Step 5-averaging vehicles' velocities with the same street ID
Step 6-calculating costs of different streets
Step 7- running Dijkstra algorithm

Intermediate nodes are used both for finding SP and for controlling traffic light. In SP finding, intermediate nodes should recognize to add their current velocity, their ID, and their street ID. But in the returning path, they act as relay. An important point about all of the above applications is that a handshaking mechanism is needed for choosing next hop. The purpose of handshaking is to choose the furthest vehicle in the transmission range of the vehicle which carries information now. In figure 1, imagine a packet in vehicle A is ready for transmission. Handshaking mechanism should be able to choose C (the furthest node) as the next relay automatically.

Step1- A inserts its ID, its current velocity, and street ID in packet.
Step 2- A sends the request packet.
Step 3- B and C receive the request packet.
Step 4- B sends an Ack which contains its location, C sends an Ack which contains its location.
Step 5- A receives B's and C's Ack.
Step 6- B and C receive each others' Acks.
Step 7- A sends an Ack which contains C's ID; C has been chosen as the next relay.
Step 8- C transmits the packet, B drop the packet.

3 Simulation

We used SUMO [6], and NS2 [7] as our simulation tools. We used SUMO to simulate semi-real traffic patterns. In this work at first, we ran SUMO to gain the traffic patterns. Based on each traffic pattern, delay of centrally controlled traffic light was calculated by SUMO. We evaluated the delay of multiple hops for each link by ns2. Then Dijkstra algorithm was used. We ran our TLC algorithm in SUMO for the same traffic pattern used in centrally controlled traffic light (centrally controlled traffic light is a traffic light without wireless capabilities).

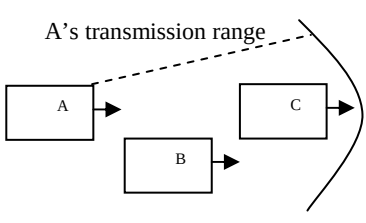


Fig. 1. MAC mechanism chooses C as the next relay

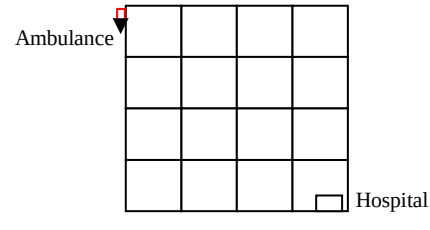


Fig. 2. Simulated urban scenario

The ambulance's source and destination is shown in figure 2. Other vehicles in the scenario have completely random movements with random source and destination. Number of vehicles is varied from 100 to10000 (0.5 veh/km to 50 veh/km). Imposed latency on travel time of the ambulance was calculated both for a centrally controlled traffic light system and for absolute priority. Simulation results have been shown in figure 3. Travelling Time Ratio is defined as: $TTR = \frac{Travellig\ \ Time\ of\ \ Ambulance}{Ideal\ Travelling\ \ Time}$

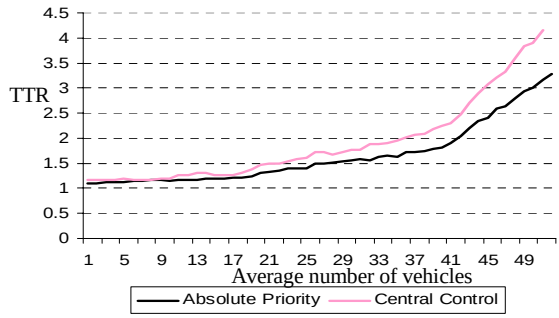


Fig. 3. Travelling Time Ratio versus average number of vehicles

Ideal Travelling Time (ITT) is the time that ambulance can travel from source to destination with maximum velocity which is 60 km/h in our simulation. ITT depends on the length of the SP. This is because increasing number of vehicles imposes upper delay. Also, by increasing average number of vehicles, the distance between these curves will increase. It shows that in rush hours, our method is more beneficial.

4 Conclusion

A decentralized protocol in VANET environment for finding SP and TLC was proposed. Simulation results show the performance improvement of our proposed protocol especially in rush hours. For future works, we try to perform an adaptive SP algorithm to address the dynamic nature of traffic patterns. Also, we want to simulate an environment which all cars announce their delay requirements. Based on these announcements, an intelligent decision is made to optimize the overall delay in a city.

Acknowledgments. This work was supported by Iran Telecommunication Research Center (ITRC).

References

1. Johnson, D., Maltz, D.: Dynamic Source Routing in Ad-Hoc Wireless Networks. In: Proc. ACM SIGCOMM (1996)
2. Bertsekas, D., Gallager, R.: Data Networks, 2nd edn., p. 401.

3. Kumar, R., Reddy, D., Singh, V.: Intelligent transport system using GIS, Map India (2003)
4. Huang, Y.-S.: Design of traffic light control systems using statecharts. The computer journal 49(6) (2006)
5. Wiering, M., Veenen, J.V., Vreeken, J., Koopman, A.: Intelligent Traffic Light Control. Institute of information and Computing and Science, Utrecht University (2004)
6. http://sumo.sourceforge.net/docs/documentation.shtml
7. http://www.isi.edu/nsnam/ns/

An Ontology Based Routing Index in Unstructured Peer-to-Peer Networks*

Hoda Mashayekhi, Fatemeh Saremi, Jafar Habibi, Habib Rostami, and Hassan AbolHassani

Computer Eng Department, Sharif University of Technology, Tehran, Iran
{mashayekhi,saremi}@ce.sharif.ir,
{jhabibi,habib,abolhassani}@sharif.ir

Abstract. We present an ontology based indexing method for unstructured P2P networks. We maintain limited size indexes at each node to identify number of documents accessible on each concept without loosing any indexing information. Out method supports complex queries, combined of conjunction and disjunction phrases. We simulate our search algorithm on an unstructured P2P network and show that our method significantly reduces the network traffic.

1 Introduction

We propose an ontology based search algorithm in unstructured P2P systems which employs local indexes to route the query through the network. By utilizing a global ontology, documents are categorized and indexed using semantic concepts. Queries are mainly forwarded on those links where the probability of finding results is noticeable. This method besides saving storage space for indexes, prevents flooding the queries in the network, and thus reduces message traffic significantly.

In early Gnutella [1] queries were flooded through the network, which lead to poor performance. Gnutella now uses distributed data indexes, with hierarchy in the form of Ultrapeers. Authors in [2] use semantic overlay networks (SONs) which consist of semantically related nodes, to improve search results. OLI [3] is an ontology based indexing method. This algorithm is most suitable for structured P2P networks as it assumes a previously available routing algorithm in the network.

2 Ontology Based Local Index Description

We assume a global ontology constructed in a tree hierarchy, available to all nodes.

Definition 1. $localOnt_u$ is a *k*-tuple index maintained in node *u*, which presents the concepts available in local database of this node. Each tuple has the form (*Cnpt*, *noD*) where *Cnpt* is one of the concepts defined in the global ontology and *noD* is the number of documents which have the subject *Cnpt* and reside in node *u*.

* This research was in part supported by a grant from Iran Telecommunication Research Center.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 960–963, 2008.

Size of the $localOnt_u$ index should be no larger than *k*. If number of distinct subjects held in node *u* exceeds *k*, they should be replaced with their least common subsumer concepts [4].

Definition 2. $linkOnt_u(i)$ is a *k*-tuple index maintained in node *u*, which represents the concepts accessible via link *i* of *u*. Each tuple has the form (*Cnpt*, *noAD*).
If node *u* is connected to node *v* via link *i*, $linkOnt_u(i)$ is the aggregation of documents located at *v* and the concepts accessible via other links of node *v*.

2.1 Query Processing

Any node receiving this query first evaluates it against its local database and retrieves the matching documents. If the stop condition is not met, it forwards the query to its neighbors sequentially. The neighbor nodes evaluate the query in a similar fashion until desired number of results is found.

Suppose the query [*q*, *n*] in which *q* is the query phrase and *n* is the requested number of results, is submitted to node *u*. If *n* documents are not discovered locally, *u* calculates the estimated number of matching documents along each of its links. Any query after expansion, will be of the regular form $A(|A)^*$, where *A* itself has the form $a(\wedge a)^*$.

Definition 3. $match_u(i,q)$ is the estimated number of results obtained by forwarding the query *q* along link *i* of node *u*.

$$match_u(i,q)=\sum_j E_i(A_j) \tag{1}$$

$$E_i(A_j)=numberOfDocuments_i \times \prod_x \frac{E_j(a_x)}{numberOfDocuments_i} \tag{2}$$

$numberOfDocuments_i$ is the number of total documents on any concept accessible via link *i* of node *u*, and *x* in the equation above iterates on the individual concepts in A_j. Let b_x be equal to a_x if a_x appears in $linkOnt_u(i)$, else the nearest successor or predecessor of a_x in the hierarchical global ontology which appears in $linkOnt_u(i)$, and if neither of the conditions exist, b_x is set to null.

$$E_j(a_x)=noAD_i\{where\ Cnpt=b_x\}\times d(b_x,a_x) \tag{3}$$

$noAD_i\{where\ Cnpt=b_x\}$ indicates the number of documents on concept b_x in $linkOnt_u(i)$. $d(b_x,a_x)$ is set to 1 if b_x is equal to a_x, or b_x is a successor of a_x. If b_x is a predecessor of a_x, consider the path between a_x and b_x, if y iterates on the elements of this path starting at parent of a_x and ending at b_x, and c_y represents the corresponding element, the value of $d(b_x,a_x)$ is determined as follows:

$$d(b_x,a_x)=\prod_y \frac{1}{number\ of\ children\ of\ c_y+1} \tag{4}$$

In sequential forwarding node u selects the link i with the highest $match_u(i,q)$ value and forwards the message $[q, n-n_r]$ on this link. After receiving the response on this link, if $n-n_r > 0$ the query is forwarded on the next link.

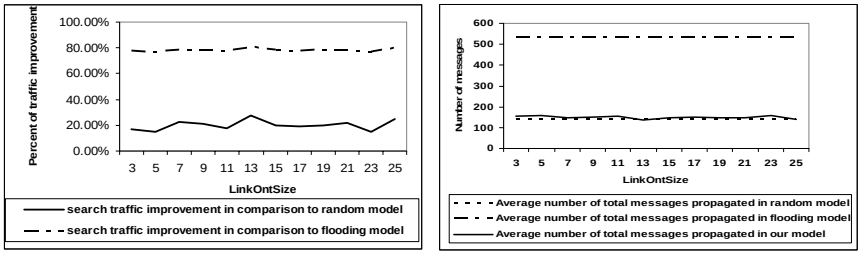
2.2 Updating and Maintaining Indexes

Definition 4. *aggOnt(v)* is a k-tuple index and is obtained by aggregating $localOnt_v$ and $linkOnt_v$ *(i)* indexes, where *i* iterates over all links of *v*.

Any change in a node's contents, or network topology results in a propagating update message which contains *aggOnt(v)*. For the aggregation, first *localOnt* and *linkOnt* indexes are put together to achieve a *(linkCnt + 1)*k-tuple* accumulated index, where *linkCnt* is the number of *v*'s links. Tuples of all *linkOnt* indexes are changed by means of multiplying the parameter *noAD* of each tuple, in a weight α to take into account the number of hops between the source and destination nodes. Achieving a *k*-tuple index is possible by finding the least common subsumers of the inputs.

3 Experimental Results

We have extended the PlanetSim simulator [5] to simulate our algorithm in an 1000 node unstructured P2P network.

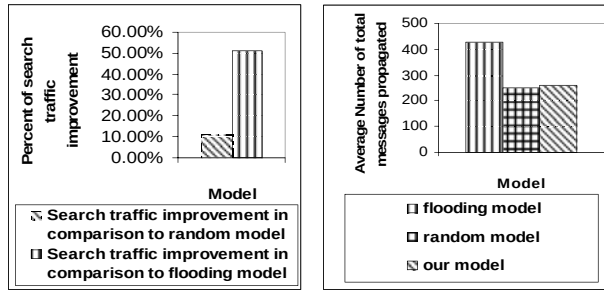


(a) (b)

Fig. 1. Effect of changing the size of link indexes on a) search traffic improvement, and b) overall traffic improvement

We consider a power law distribution of concepts and number of neighbors. We have compared our model to two other models. In the flooding model, each node after locally processing the query, forwards it on all of its links, except the input link while in the random model, the query is forwarded along a random link. Fig. 1 shows the effect of changing index size of each link (*linkOntSize*), on search traffic improvement.

With respect to design of our algorithm, good performance is expected when queries contain 'and' phrases. Fig. 2 (a) shows improvement in search traffic when approximately 80% of the queries are 'and' queries. Fig. 2 (b) compares the overall traffic of three models in this configuration. As observed, while the search cost is improved efficiently, there is no extra cost in implementing our algorithm.



(a) (b)

Fig. 2. Results regarding 'and' queries, (a) search traffic, (b) overall traffic

4 Conclusion

In our ontology based indexing method for unstructured P2P systems, each node maintains *k*-tuple indexes for its local database and each outgoing link. Storage space, accuracy of requested results and response time in the search process is significantly improved by our proposed algorithm.

References

[1] Gnutella, http://www.gnutella.com
[2] Crespo, A., Garcia-Molina, H.: Semantic Overlay Networks for P2P Systems. In: Moro, G., Bergamaschi, S., Aberer, K. (eds.) AP2PC 2004. LNCS (LNAI), vol. 3601, pp. 1–13. Springer, Heidelberg (2005)
[3] Rostami, H., Habibi, J., Abolhassani, H., Amirkhani, M., Rahnama, A.: An Ontology Based Local Index in P2P Networks. In: SKG 2006 conference (2006)
[4] Cohen, W., Borgida, A., Hirsh, H.: Computing least common sub-sumer in description logics. In: Swartout, W. (ed.) Proc. 10th Nat Conference on Artificial Intelligence (AAAI 1992), San Jose, CA, pp. 754–760. MIT Press, Cambridge (1992)
[5] García, P., Pairot, C., Mondéjar, R., Pujol, J., Tejedor, H., Rallo, R.: PlanetSim: A new overlay network simulation framework. In: Gschwind, T., Mascolo, C. (eds.) SEM 2004. LNCS, vol. 3437, pp. 123–137. Springer, Heidelberg (2005)

Two Architectural Practices for Extreme Programming

Amir Azim Sharifloo, Amir S. Saffarian, and Fereidoon Shams

Electrical and Computer Engineering Dept.,
Shahid Beheshti University,
Evin, Tehran, Iran
a.sharifloo@mail.sbu.ac.ir,
a_saffarian@std.sbu.ac.ir,
f_shams@sbu.ac.ir

Abstract. Extreme programming (XP) is one of the important agile methods and is being used widely. Although XP itself is quite new, many of the practices utilized by it have been around for some time. The most controversial aspect of XP is the change management aspect of the process. In XP, the on-site customer requests the changes informally, often by verbally informing the development team. In spite of all benefits known about XP, it does not involve architectural techniques that are necessary to expect acceptable level of quality for the system being developed. This paper introduces two practices in order to empower XP's development process toward improving system's architecture. The main characteristic of proposed solution is that it is derived from values and practices of XP so as to be compatible with this process model and to keep its agility intact.

Keywords: Extreme Programming, Agile, Architecture, Refactoring.

1 Introduction

In the last decade, agile methods are emerged as a novel and efficient approach to develop software systems. Most important purpose of them is customer satisfaction, responding to requested changes during software development and finally achieving suitable profits. Agile methods, unlike conventional approaches, try to solve problems that derive from project time constraints. In order to do that, they are based on people skills and their interactions to each other that could accelerate the process of system development. They do try to appropriately communicate with customer to reduce iteration`s period, and resolving problems such as low flexibility and weakness on satisfying customer's requirements.

Meanwhile, several agile methods have been presented that are accepted and applied to software engineering community like XP, FDD, and SCRUM. Among them, XP is one of the most popular methods currently being used and different kinds of research have been accomplished about its various aspects like pair programming [1, 2, and 3]. XP method was created around four values: communication, simplicity, feedback, and courage. Various principles, practices, and roles are derived based on these values that through one process model establish XP's foundation [4].

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 964–967, 2008.

In XP, most of the design activities are performed by programmers through pair programming practice. During software development process, design of system could suffer from a lot of modifications. Additionally, because of distributed nature of the development process, final product may have a weak structure. Therefore, even if final product satisfies all functional requirements, it does not take into account quality attributes. In this paper, the primary goal is to satisfy quality attributes when developing a system using XP method. We introduce two Architectural Practices Continuous Architectural Refactoring (CAR) and Real Architecture Qualification (RAQ) that are applied in XP concurrently with other practices. In the following, section 2 describes the background. CAR and RAQ are illustrated in section 3.

2 Background

Today, Software architecture focuses on achieving quality attributes for a software system. Therefore using software architecture skills in XP could improve it toward producing a software system that has an appropriate structure beside an acceptable quality level. Cockburn characterizes XP method as being appropriate for projects with small size team that have a criticality where defects won't cause loss of life. For larger projects, he suggests to use the architect role [5]. Paul McMahon tried to extend agile approaches to be used in large, distributed projects by considering usage of architectural practices [6]. At the time of writing this paper we have found only one reported work but it has some shortcomings [7, 8]. They tried to integrate software architecture techniques with XP. But this kind of integration is not applicable into a real XP team because of the fact that they are not derived from XP values and practices and are not in the way of agile principles.

In [7], the authors' focus is on integrating software architecture techniques with XP. These techniques are QAW, ADD and ATAM/CBAM which have been presented by SEI[1]. QAW is an architectural workshop that is held to reach better elicitation of system's quality attributes [9]. In their work, QAW is held on first iteration of XP and help enrich user stories by adding quality attribute scenarios and consequently improves requirement elicitation. In this integration, ADD [10] is used to design an overall architecture that is considered as a basic system structure. ATAM/CBAM is a combined architecture evaluation method of ATAM and CBAM [11] that is used in the reported work. ATAM does an analysis on frequent outputs of development teams and CBAM considers next iteration decisions based on cost and benefits.

There are some shortcomings that make mentioned approach a non-practical integration. Results of this enrichment is an increase in the amount of documents and artifacts and the time that its session takes could be consider a sensible portion of the total project time.

3 Two Architectural Practices

Quality attributes could be divided into two categories considering the perspective to be used. The first category is related to the quality attributes that are evaluated based on

[1] Software Engineering Institute.

developers' concerns like modifiability, testability, and etc. Ordinarily, this evaluation is done by architect, who has knowledge about architectural design, through system models and diagrams in the CAR process. The other category contains quality attributes which should be evaluated using the real working system like performance, usability, availability. These quality attributes are observable externally considering users view. Therefore, customer is the person to evaluate these qualities through RAQ.

3.1 CAR Practice

The main role of this practice is architect who has a good knowledge about architectural and design decisions and architectural smells too. She/he receives architectural models developed by pair programming teams and integrates them as a partial architectural diagram. When model integration occurs, the architect will do a thorough analysis on the integrated diagram to identify smells and provide solutions for them as new tasks. New tasks are considered as Unfinished Tasks [12] that, if possible, are accomplished in current iteration or could be planed to be done in following ones.

Note that the analysis process of CAR is essentially quality driven. Some of quality attributes that are suitable to be analyzed using CAR are modifiability, testability and scalability that are important for developers like programmers. The CAR method is a really beneficial process that empowers XP from architectural aspects but it is not enough because (1) it is done only by the architect, not taking others` opinion into account and (2) not all quality attributes could be analyzed efficiently in the CAR practice like performance. So to provide a complementary solution that could cover those problems, we introduce RAQ practice.

3.2 RAQ Method

Real Architecture Qualification is almost a kind of brainstorming session based on a working system and architectural models that embodies certain steps. RAQ is a practice that becomes active in the end of all iterations. Its main goal is to test the working system that outcomes an iteration according to customer requirements. Steps of RAQ could be summarized as:

1. Preparing the structure of the session and identifying representatives of stakeholders to join the session.
2. Describing, in brief, final architectural model of the iteration that is integrated using partial models.
3. Describing, in brief, refactoring decisions that have be made by architect and ask members to express their opinions about them.
4. Identifying architectural aspects (quality attributes) that should be analyzed in the session.
5. For each quality attribute specified in step 4:
   - Start a brainstorming sub-session, taking stakeholder`s opinions into account about their experience when working with current working system.
   - Decide about solutions for new needs and requests that current working system cannot provide or handle.
   - Defining new solutions as concrete new tasks and taking them as unfinished tasks.

References

1. Williams, L., Kessler, R.R., Cunningham, W., Jeffries, R.: Strengthening the Case for Pair Programming. IEEE Software 17(4), 19–25 (2000)
2. Arisholm, E., Gallis, H., Dybå, T., Sjøberg, D.I.K.: Evaluating Pair Programming with Respect to System Complexity and Programmer Expertise. IEEE Transactions on Software Engineering 33(2), 65–86 (2007)
3. Vanhanen, J., Abrahamsson, P.: Perceived Effects of Pair Programming in an Industrial Context. In: 33rd EUROMICRO Conference on Software Engineering and Advanced Applications (EUROMICRO 2007), pp. 211–218 (2007)
4. Beck, K., Andres, C.: Extreme Programming Explained: Embrace Change, 2nd edn. Addison-Wesley Professional, Reading (2004)
5. Cockburn, A.: Agile Software Development, 1st edn. Addison-Wesley Professional, Reading (2001)
6. McConnell, S.C.: Rapid Development: Taming Wild Software Schedule. Microsoft Press (1996)
7. Nord, R., Tomayko, J.: Software Architecture-Centric Methods and Agile Development. IEEE Software 23(2), 47–53 (2006)
8. Nord, R., Tomayko, J., Wojcik, R.: Integrating Software-Architecture-Centric Methods into Extreme Programming (XP). Technical Note, CMU/SEI-2004-TN-036, Software Engineering Institute, Carnegie Mellon University (2004)
9. Barbacci, M.R., Ellison, R., Lattanze, A.J., Stafford, J.A., Weinstock, C.B., Wood, W.G.: Quality Attribute Workshops (QAWs). Technical Report, 3rd Edn., CMU/SEI-2003-TR-016, Software Engineering Institute, Carnegie Mellon University (2003)
10. Wojcik, R., Bachmann, F., Bass, L., Clements, P., Merson, P., Nord, R., Wood, B.: Attribute-Driven Design (ADD). Technical Report, Version 2.0, CMU/SEI-2006-TR-023, Software Engineering Institute, Carnegie Mellon University (2006)
11. Nord, R., Barbacci, M., Clements, P., Kazman, R., O'Brien, L., Tomayko, J.: Integrating the Architecture Tradeoff Analysis Method(ATAM) with the Cost Benefit Analysis Method (CBAM). Technical Note, CMU/SEI-2003-TN-038, Software Engineering Institute, Carnegie Mellon University (2003)
12. Arisholm, E., Gallis, H., Dybå, T., Sjøberg, D.I.K.: Evaluating Pair Programming with Respect to System Complexity and Programmer Expertise. IEEE Transactions on Software Engineering 33(2), 65–86 (2007)

Adaptive Target Detection in Sensor Networks

Ghasem Mirjalily

Electrical Engineering Department
Yazd University, Yazd, Iran
mirjalily@yazduni.ac.ir

Abstract. In a sensor detection network, each sensor makes a binary decision based on its own observation, and then communicates its binary decision to a fusion center, where the final decision is made. To implement an optimal fusion center, the performance of each sensor as well as the a priori probabilities of the hypotheses must be known. However, these statistics are usually unknown or may vary with time. In this paper, I will introduce a recursive algorithm that approximates these values on-line and adapts the fusion center. This approach is based on time-averaging of the local decisions and using them to estimate the error probabilities and a priori probabilities of the hypotheses. This method is efficient and its asymptotic convergence is guaranteed.

Keywords: Target detection, Sensor networks, Stochastic approximation.

1 Introduction

Recently, sensor networks have attracted much attention and interest, and have become a very active research area [1]. Sensor networks have wide applications and high potential in target detection, military surveillance, security, monitoring of traffic, and environment. In this paper, we focus on distributed target detection, one of the most important functions that a sensor network needs to perform. Usually, a sensor network consists of a large number of low-cost and low-power sensors, which are deployed in the environment to collect observations and preprocess the observations. Each sensor node has limited communication capability that allows it to communicate via a wireless channel. Normally, there is a fusion center that processes data from sensors and forms a global situational assessment [2].

In this topology, for target detection, *i*-th sensor (S_i) makes a decision (u_i) based on its observation (Y_i) and then transmits this decision to the fusion center. Based on the received decisions, the fusion center makes the final decision (u_0) by employing the fusion rule.

In target detection problem, we have binary hypothesis testing with two hypotheses H_0 and H_1, therefore u_i will be binary. When the *i*-th sensor favors the H_1 hypothesis, $u_i = 1$; otherwise $u_i = 0$. The *a priori* probabilities of the two hypotheses are denoted by $P(H_0)=P_0$ and $P(H_1)=P_1$ such that $P_0 + P_1 = 1$. Chair and Varshney [3] showed that for N conditionally independent sensors, the optimal fusion rule based on the minimum probability of error criterion is:

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 968–971, 2008.

$$u_0 = \begin{cases} 1, & if \ w_0 + \sum_{j=1}^{N} w_j > 0 \\ 0, & Otherwise, \end{cases} . \tag{1}$$

Where

$$w_0 = \log(\frac{P_1}{P_0}), \ w_j = \begin{cases} \log(\frac{1-\beta_j}{\alpha_j}), & if \ u_j = 1 \\ \log(\frac{\beta_j}{1-\alpha_j}), & if \ u_j = 0. \end{cases} \quad j=1,2,\ldots,N, \tag{2}$$

Here α_j and β_j represent the probabilities of *first kind error (false alarm)* and *second kind error (missed detection)* at the *j*th sensor. So

$$\alpha_j = \Pr(u_j = 1 | H_0), \ \beta_j = \Pr(u_j = 0 | H_1). \tag{3}$$

As can be seen, each weight w_j is a function of the error probabilities at the corresponding sensor. So to implement the optimal fusion rule, the *a priori* probabilities of the hypotheses and the error probabilities corresponding to each sensor must be known. Unfortunately these statistics are not always available in practice or may be time-varying. In our previous work, we have developed an algorithm for estimating these values on-line at the fusion center for distributed radar detection systems [4]. The primary objective of this work is to introduce our algorithm again for deploying it in sensor detection networks.

2 Adaptive Estimation Algorithm

First, we introduce an adaptive estimation algorithm for the simple case with three sensors. Later, this algorithm will be generalized for systems with more than three sensors. Note that our algorithm is not applicable in systems with two sensors. Consider a sensor detection network with only three sensors. Our propose is to estimate the unknown probabilities P_0, P_1, α_i and β_i $(i = 1,2,3)$ by observing the binary decisions u_1, u_2 and u_3. Let us define some notations

$$\gamma_i = \Pr(u_i = 1), \ \ \delta_{ij} = \Pr(u_i = 1, u_j = 1), \ \ \eta = \Pr(u_1 = 1, u_2 = 1, u_3 = 1). \tag{4}$$

Using these definitions, we can write

$$\gamma_i = (1-\beta_i)P_1 + \alpha_i(1-P_1), \ i = 1,2,3 \tag{5}$$

$$\delta_{ij} = (1-\beta_i)(1-\beta_j)P_1 + \alpha_i \alpha_j (1-P_1), \ \ i, j = 1,2,3, \ j \neq i \tag{6}$$

$$\eta = (1-\beta_1)(1-\beta_2)(1-\beta_3)P_1 + \alpha_1\alpha_2\alpha_3(1-P_1) \tag{7}$$

The above set of seven equations can be solved simultaneously and we get an unique solution for P_1 and P_0:

$$P_1 = 0.5 - \frac{X}{2\sqrt{X^2+4}},\ P_0 = 1 - P_1, \tag{8}$$

Where

$$X = \frac{(\eta + 2\gamma_1\gamma_2\gamma_3) - (\gamma_1\delta_{23} + \gamma_2\delta_{13} + \gamma_3\delta_{12})}{\sqrt{K_{12}K_{13}K_{23}}},\ K_{ij} = \delta_{ij} - \gamma_i\gamma_j. \tag{9}$$

The other six unknown probabilities can also be calculated explicitly

$$\alpha_i = \gamma_i - \sqrt{\frac{P_1}{1-P_1}}\,a_i,\ \ \beta_i = 1 - \gamma_i - \sqrt{\frac{1-P_1}{P_1}}\,a_i \quad i = 1,2,3. \tag{10}$$

where P_1 is given by (8) and

$$a_1 = \sqrt{\frac{K_{12}K_{13}}{K_{23}}},\ a_2 = \sqrt{\frac{K_{12}K_{23}}{K_{13}}},\ a_3 = \sqrt{\frac{K_{13}K_{23}}{K_{12}}}. \tag{11}$$

Therefore all of the unknown probabilities P_0, P_1 and α_i, β_i $(i = 1,2,3)$ can be calculated uniquely from unconditional probabilities $\gamma_i\,(i = 1,2,3)$, $\delta_{ij}\,(i = 1,2,3,\ j \neq i)$ and η. Notice that the values of these unconditional probabilities can be conveniently approximated via time-averaging [5], e.g.,

$$\hat{\gamma}_i^k = \frac{1}{k}\sum_{j=1}^{k} u_i^j, \tag{12}$$

where u_i^j is the decision of *i*th sensor at the *j*th time step. The above equation can be written recursively as

$$\hat{\gamma}_i^k = \frac{1}{k}u_i^k + \frac{k-1}{k}\hat{\gamma}_i^{k-1}, \tag{13}$$

with an arbitrary initial guess. Similar to (13), we have

$$\hat{\delta}_{ij}^k = \frac{1}{k}u_i^k u_j^k + \frac{k-1}{k}\hat{\delta}_{ij}^{k-1},\ \ \hat{\eta}^k = \frac{1}{k}u_1^k u_2^k u_3^k + \frac{k-1}{k}\hat{\eta}^{k-1}. \tag{14}$$

Once the estimated values $\hat{\gamma}_i^k, \hat{\delta}_{ij}^k$ and $\hat{\eta}$ are obtained, all of the unknowns P_0, P_1 and α_i, β_i can be estimated by substituting these estimated values for γ_i, δ_{ij} and η in equations (8)-(11). Equations (12)-(14) are stochastic approximations which are convergent with probability 1 [5].

Now consider the problem of adaptive estimation for a sensor detection network with N sensors $(N > 3)$. In this case, we have *2N+1* unknown to be estimated: α_i, β_i $(i = 1,2,...,N)$ and P_1. From equations (5) and (6), we have

$$\alpha_j = \frac{(P_1\delta_{ij} - \gamma_i\gamma_j)/(1-P_1) + \gamma_j\alpha_i}{-\gamma_i + \alpha_i}, \quad \beta_j = 1 + \frac{\frac{-(1-P_1)\delta_{ij} + \gamma_i\gamma_j}{P_1} - \gamma_j(1-\beta_i)}{1-\gamma_i-\beta_i} \tag{15}$$

This equation shows a relationship between the probabilities α_i and β_i of *i*th sensor and the probabilities α_j and β_j of *j*th sensor. Thus, if the probabilities α_i and β_i have estimated for *i*th sensor, we can use the equation (15) to obtain estimates of probabilities α_j and β_j for any other sensor *j*. This observation suggests that we can arbitrarily select any set of three sensors, say S_1, S_2 and S_3, and use the adaptive estimation algorithm on these sensors. As a result, we obtain estimates of P_1, α_i and β_i for these three sensors. Next, we select one of the three sensors as the *reference sensor*, say S_1. Then by using (15) and the probabilities α_1 and β_1 corresponding to reference sensor S_1, we can calculate the estimates for α_j and β_j $(j = 4,5,...,N)$. In this estimation process, we need to estimate (2N+1) unconditional joint probabilities: γ_i $(i = 1,2,... ,N)$, δ_{1j} $(j = 2,..., N)$, δ_{23} and η.

3 Conclusion

In a sensor detection network, the performance of sensors may be unknown or variable. Under such circumstances, we need an adaptive fusion center in order to obtain optimal performance. In this paper, I introduced a recursive algorithm based on time-averaging of the local decisions and using them to estimate the error probabilities of the sensors and a priori probabilities. This algorithm is based on an explicit analytic solution and is suitable for a time-varying environment.

References

1. Sohrabi, K., Minoli, D., Znati, T.: Wireless Sensor Networks. John Wiley, Chichester (2007)
2. Niu, R., Varshney, P.K.: Distributed Detection and Fusion in a Large Wireless Sensor Network of Random Size. EurAsiP Journal on Wireless Communications and Networking 4, 462–472 (2005)
3. Chair, Z., Varshney, P.K.: Optimum Data Fusion in Multiple Sensor Detection Systems. IEEE Trans. Aerospace Elect. Syst. 22(1), 98–101 (1986)
4. Mirjalily, G., Luo, Z.Q., Davidson, T.N., Bose, E.: Blind Adaptive Decision Fusion for Distributed Detection. IEEE Trans. Aerospace Elect. Syst. 39(1), 34–52 (2003)
5. Robbins, H., Monro, S.: A Stochastic Approximation Method. Annals of mathematical statistics 22, 400–407 (1951)

Linear Temporal Logic of Constraint Automata

Sara Navidpour[1] and Mohammad Izadi[2]

[1] Department of Mathematical Science, Sharif University of Technology, Tehran, Iran
Sara.Navidpour@gmail.com
[2] Research Institute for Humanities (IHCS) and Department of Computer Engineering, Sharif University of Technology, Tehran, Iran
izadi@ce.sharif.edu

Abstract. Constraint automata are formalisms to describe the behavior and possible data flow in coordination models. In this paper we introduce a linear time temporal logic, called temporal logic of steps (TLS), for specifying the executions of constraint automata. TLS is the first attempt in presenting a linear temporal logic for constraint automata. Having TLS in hand, we are able to design model checking algorithms for verification of concurrent systems modeled by constraint automata.

Keywords: Constraint Automata; Concurrent System; Temporal Logic; Coordination Model, Weak and Strong Fairness.

1 Introduction

Temporal logics have proved to be useful for specifying concurrent systems, because they can describe ordering of events in time without explicitly introducing time. They are often classified into linear time and branching time temporal logic categories. They have different expressive powers. An important example is that, fairness properties can not be expressed directly in branching time logic CTL but they can be described in temporal logics with linear operators such as LTL [3]. In this paper we introduce a linear time temporal logic, which we call temporal logic of steps (TLS), for specifying executions in constraint automata. Constraint automaton is a formalism to describe the behavior and data flow in coordination models [2].

1.1 Basic Theory of Constraint Automata

A constraint automaton [2], uses a set of names, e.g., $Names = \{A_1, \ldots, A_n\}$ where each A_i represents an input/output port of a connector or component. To formulize the input/output behavior of a coordination model, constraint automata associate a timed data stream [1] to each port A_i. A timed data stream $\langle \alpha, a \rangle$ consists of a data stream $\alpha \in Data^{\omega}$ and a time stream $a \in \Re_{+}^{\omega}$, consisting of increasing positive real numbers that go to infinity. The time stream a indicates, for each data item $\alpha(k)$,

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 972–975, 2008.

the moment $a(k)$ at which it is being input or output. In [2] it's assumed that in each port there is an infinite data flow. For now, we take the same assumption, and it means that we only consider infinite behaviors of the model.

Definition 1 (*Data assignment*). Data assignment is a function $\delta : N \to Data$ where $\varnothing \neq N \subseteq Names$, and δ assigns to any port name $A \in N$ the value $\delta_A \in Data$.

The transitions of constraint automata are labeled with pairs consisting of a non-empty set N of $Names$ and a data constraint g. Formally, data constraints are propositional formulas given with the following grammar:

$$g ::= true \mid d_A = d \mid g_1 \vee g_2 \mid \neg g \quad . \tag{1}$$

Where $A \in Names$ is a name and $d \in Data$ is a data value. $DC(N, Data)$ denotes the set of data constraints that use only atoms "$d_A = d$" for $A \in N$. The Boolean connectors $\wedge$ (conjunction), $\to$ (implication), $\leftrightarrow$ (equivalence), and so on, can be derived as usual. The symbol $\models$ stands for the satisfaction relation which results from interpreting data constraints over data assignments. For instance, if $d_1 \neq d_2$

$$[A \mapsto d_1, B \mapsto d_2, C \mapsto d_1] \not\models d_A = d_B \quad . \tag{2}$$

Satisfiability and validity, logical equivalence $\equiv$, and logical implication $\leq$ (or $\models$) of data constraints are defined as usual.

Definition 2 (*Constraint Automata*). A constraint automaton A over a data domain $Data$ is a 4-tuple $(Q, Names, \to, Q_0)$, where Q is a set of states, $Names$ is a finite set of port names, $\to$ is a subset of $Q \times 2^{Names} \times DC \times Q$, called the transition relation A, $Q_0 \subseteq Q$ is a set of initial states. $(q, N, g, p) \in \to$ is often represented as $q \xrightarrow{N,g} p$. For every $q \xrightarrow{N,g} p$ it's required that (1) $N \neq \varnothing$, and (2) $g \in DC(N, Data)$.

Definition 3 (*Execution and Step*). An execution of a constraint automaton A is an infinite sequence $q_0(N_1, g_1)q_1(N_2, g_2)q_2 \ldots$ of alternating states and constraint pairs (N_i, g_i) beginning with a state, such that for all i, $q_i \xrightarrow{N_{i+1}, g_{i+1}} q_{i+1}$. Every 3-tuple $(q, (N, g), q')$ such that $q \xrightarrow{N,g} q'$ is called a step.

2 Temporal Logic of Steps

The basic ingredients of temporal logic of steps are step predicates. A step predicate is maid of conjunction of three parts. The first two parts are state predicates that depend

on state variables V in a given state, and their value in the successor state (referred to by V'). The third part which is called constraint predicate, describes data constraint of the step. A constraint predicate is a propositional formula $\varphi \wedge g$, where $g \in DC(N, Data)$ and φ is a formula that describes $N \subseteq Names$ such that:

$$\varphi = \bigwedge_{i=1}^{n} \varphi_i \quad , \quad \varphi_i = \begin{cases} A_i & if \quad A_i \in N \\ \neg A_i & if \quad A_i \notin N \end{cases} . \tag{3}$$

Thus, a step predicate is a propositional formula of the form:

$$P = P_V \wedge (\varphi \wedge g) \wedge P_{V'} . \tag{4}$$

The semantics of a step predicate P in postfix notation, and under assignment $(\sigma, \delta, \nu, \tau)$, is given as:

$$\begin{gathered} (\sigma, \delta, \nu, \tau)[\![P]\!] \quad \equiv \\ P\left(\forall v . \sigma[\![v]\!]/v \ , \ \forall v' . \tau[\![v]\!]/v' , \ \forall d_{A_i} . \delta[\![d_{A_i}]\!]/d_{A_i} \ , \ \forall A_i . \nu[\![A_i]\!]/A_i\right) \end{gathered} \tag{5}$$

$$(\sigma, \delta, \nu, \tau) \models P \quad \equiv \quad (\sigma, \delta, \nu, \tau)[\![P]\!] = true \ . \tag{6}$$

Where δ is the previously defined data assignment function, and ν assigns every variable $A_i \in N$ to *true* and every variable $A_i \notin N$ to *false*. σ and τ are assignments over state variables, each of which specifies a particular state.

A step predicate P is said to be true in a step $(q, (N, g), q')$, written as $(q, (N, g), q') \models P$, iff P_V is true in state q, $P_{V'}$ is true in state q', $g \models P_g$, and $\varphi \models P_N$. Note that P_i for $i = \{V, V', g, N\}$ is in fact the limitation of P to $\{V, V', DC(N, Data), N\}$.

2.1 Temporal Formulas of TLS

Let $\alpha = q_0(N_1, g_1)q_1(N_2, g_2)q_2 \ldots$ be an execution of constraint automaton A. Every step predicate S is a temporal formula. Furthermore, given any temporal formulas P and Q of TLS, the formulas $P \wedge Q$, $\neg P$, $\Box P$, and $\bigcirc P$ are temporal formulas of TLS as well, and the evaluation $\models$ is defined inductively as follows:

$$\begin{aligned} &\text{Step Predicate:} && \alpha \models S && \equiv \ (q_0, (N_1, g_1), q_1) \models S \ . \\ &\text{Conjunction:} && \alpha \models P \wedge Q && \equiv \ \alpha \models P \ \text{ and } \ \alpha \models Q \\ &\text{Negation:} && \alpha \models \neg P && \equiv \ \neg(\alpha \models P) \\ &\text{Always:} && \alpha \models \Box P && \equiv \ \forall_i \, . \, {}_i|\alpha \models P \\ &\text{Next:} && \alpha \models \bigcirc P && \equiv \ {}_1|\alpha \models P \ . \end{aligned} \tag{7}$$

While ${}_i|\alpha$, denotes the *i*-th suffix of α, i.e. $q_i(N_{i+1}, g_{i+1})q_{i+1} \ldots$. Given temporal formulas P and Q, the following formulas can be derived:

$$\begin{aligned} &\text{Boolean:} && P \Rightarrow Q && \equiv \neg\left(P \wedge \neg Q\right) \\ & && P \vee Q && \equiv \left(\neg P\right) \Rightarrow Q \\ &\text{Eventually:} && \Diamond P && \equiv \neg\Box\neg P \\ &\text{Leads to:} && P \rightsquigarrow Q && \equiv \Box\left(P \Rightarrow \left(\Diamond Q\right)\right) . \end{aligned} \tag{8}$$

Definition 6 (*validity*). Let A be a constraint automaton. A temporal formula P is said to be valid for A, written $A \models P$, iff P holds for every execution $\alpha \in execs(A)$. Thus,

$$A \models P \quad \equiv \quad \forall \alpha \in execs(A).\alpha \models P \ . \tag{9}$$

Having TLS in hand, we are able to describe fairness properties for constraint automata by temporal formulas, which is more convenient than the fairness sets in standard automata theory. For weak and strong fairness conditions W and S :

$$WF_A(W) \equiv \Diamond\Box Enabled(W) \Rightarrow \Box\Diamond W \ , \tag{10}$$

$$SF_A(S) \equiv \Box\Diamond Enabled(S) \Rightarrow \Box\Diamond S \ . \tag{11}$$

3 Conclusion and Future Work

A number of temporal logics for program verification of different models have already been proposed, e.g., linear temporal logic of Pnueli, CTL and CTL* [3], Temporal Logic of Actions (TLA) [4] and the Temporal Logic of Steps developed for I/O automata in [5]. This work is the first attempt to develop a linear temporal logic for constraint automata. Using TLS makes it possible to develop model checking algorithms for constraint automata.

References

[1] Arbab, F., Rutten, J.J.M.M.: A coinductive calculus of component connectors. In: Wirsing, M., Pattinson, D., Hennicker, R. (eds.) WADT 2002. LNCS, vol. 2755, pp. 34–55. Springer, Heidelberg (2003)

[2] Baier, C., Sirjani, M., Arbab, F., Rutten, J.: Modeling Component Connectors in Reo by Constraint Automata. Science of Computer Programming 61, 75–113 (2006)

[3] Clarke, E.M., Burch, J.R., Grumberg, O., Pled, D.A.: Model Checking. MIT Press, Cambridge

[4] Lamport, L.: The Temporal Logic of Actions. ACM Transactions on Programming Languages and Systems 16(3), 872–923 (1994)

[5] Muler, O.: A Verification Environment for I/O Automata -Part I: Temporal Logic and Abstraction-. TUM-INFO-06-I9911-50/1.FI (1999)

Using Social Annotations for Search Results Clustering

Sadegh Aliakbary, Mahdy Khayyamian, and Hassan Abolhassani

Department of Computer Engineering, Sharif University of Technology, Tehran, Iran
{aliakbary,khayyamian}@ce.sharif.edu,
abolhassani@sharif.ir

Abstract. Clustering search results helps the user to overview returned results and to focus on the desired clusters. Most of search result clustering methods use title, URL and snippets returned by a search engine as the source of information for creating the clusters. In this paper we propose a new method for search results clustering (SRC) which uses social annotations as the main source of information about web pages. Social annotations are high-level descriptions for web pages and as the experiments show, clustering based on social annotations yields good clusters with informative labels.

Keywords: Social Annotation, Clustering, Search Engine, Tagging.

1 Introduction

Today, search engines are the most popular services on the web and they have become the gateway of many users to the internet. Search engines refine their results in many ways in order to help the user find target information easily. Query expansion, query recommendation and spell correction are some classic search results refinements offered by search engines. Another refinement which some search engines (like Vivisimo [12] and Carrot [13]) offer is search results clustering (SRC). The goal of SRC is grouping search results into some dynamic categories and naming the categories with informative labels, providing a better view of search results so that the user can rapidly find related information. SRC would be helpful especially when query is vague, too broad or simply ill-defined [4].

The problem of SRC is investigated by many researchers, most of them (like [1,4,11,15]) have used title, URL and snippets(excerpts) returned by a search engine as the source of data. This is because by using actual page contents, the cluster construction would be significantly slow. We will use another source of information which helps to construct better clusters with better labels: “Social annotations”. Social annotations - also referred to as “collaborative tags” - are simple keywords assigned to URLs by ordinary web users. This is the practice of allowing anyone to freely attach keywords or tags to content [2]. These tags, when shared and aggregated, bring a semi-taxonomy called “folksonomy”. Today, many online services e.g. Delicious [9], YahooMyWeb [10], Diigo [21] and Technorati [14] have been developed for web users to organize and share their favorite web pages online, using social annotations.

Social annotations help us to construct meaningful categories of search results and good category labels, with faster algorithms. Traditional algorithms, which use

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 976–980, 2008.

snippets for creating clusters, have some drawbacks: snippets are usually correlated with the query and sometimes do not reflect the content of the corresponding web page correctly. On the other hand social annotations are high level descriptions for web pages and we can effectively use them for SRC.

Some related works in this area are [1,4,11,15]. Reference [1] uses a supervised learning approach for creating clusters. Reference [4] first identifies some cluster labels as the cluster representatives and then assigns the documents to these labels. The famous Vivisimo web site [12] uses an algorithm similar to Lingo, but in addition to text snippets (returned by msn live [17]) it uses open directory of dmoz [18] and Ask web site [16]. References [3,5,6,7,8] have used social annotations for problems previously attacked with other approaches. Social annotation services are still immature and research on these bookmarks is recently started.

2 Proposed Method

Our proposed method uses social annotations. It sends query to Google search engine and then the returned results are processed: the tags assigned to each returned URL are retrieved from tag database, and results are clustered using these tags. For each retrieved URL we create a document containing tags in Delicious web site [9] for the URL. This document is represented by a vector in text vector space representation model. These vectors, called "tag vectors", are clustered using vector-based similarity measures. In fact instead of clustering page contents or the snippets, the tag sets corresponding to pages are clustered. For labeling each cluster the tags in the cluster are analyzed and most descriptive tags are chosen for the cluster label.

Our method for SRC is implemented in four phases: data gathering, preprocessing, cluster construction and cluster labeling.

Data Gathering: Delicious web site [9] is used as the source of social annotations. We explored Delicious automatically using a crawler application and gathered a database of tags to be used in three following phases. The database contains URLs, their corresponding tags and frequency of each tag for each URL. In order to avoid mistakenly created tags, the tags with the frequency equal to one are not saved.

Preprocessing: In preprocessing phase tag database is refined. Tag refinement is done in two stages: "bad tags removal" and "root analysis". In fact some tags, like "myStuff", "toRead", "other" and "interesting" are only informative for the tagger himself/herself. These kinds of tags, which are useless in process of cluster construction, are categorized as "Self Reference" and "Task Organizing" tags in [2]. There are also tags like "safari_export" and "imported_ie_favorites" that are generated automatically by tagging services. We maintained a black-list and filled it with these inapplicable tags, gathered manually by watching delicious popular tags in our database. These tags are removed from tag database in "bad tag removal" stage.

In root analysis stage some tags with similar roots are merged according to a simple stemming task. The merging is done for tags of each URL according to their frequency. For example http://espn.go.com is tagged both with "sport" and "sports" tags, but frequency of "sports" tag is more than frequency of "sport" tag, so we regarded all "sport" tags as "sports" for this URL. In fact all tags with the same root are automatically replaced with the most frequent one for each URL. As a result, similar tag pairs

like <shopping, shop>, <e-mail, email> and <source, sources> are merged for most of URLs, but a "news" tag is never transformed to "new". The Porter algorithm [19] is used to automatically find tags with similar roots for merging.

Cluster Construction: In cluster construction phase, search results are clustered using their corresponding tags. We used Google search engine for gathering search results: the query is sent to Google and the first 100 results are retrieved.

We examined some hierarchical clustering algorithms and also k-means [20] algorithm, but finally we chose k-means because of its better results. Input of clustering algorithm is a set of URLs, their corresponding tags and tag frequency for each <URL, tag> pair. We regard each page returned by search engine as a vector of tags with frequency of the tag as its magnitude in the vector. For example a URL with 50 "sport" tags, 30 "news" tags and 10 "football" tags, is regarded as the vector: <(sport, 50) , (news, 30) , (football,20)> which we call it the "tag vector" of the URL. Cosine similarity measure is used for calculating similarity between two URLs:

$$\cos(a,b) = \frac{\vec{a} \bullet \vec{b}}{|\vec{a}| \cdot |\vec{b}|} = \frac{\sum_{i=1}^{t} (a_i \cdot b_i)}{\sqrt{\sum_{i=1}^{t} a_i^{\,2} \cdot \sum_{i=1}^{t} b_i^{\,2}}} \tag{1}$$

where **a** and **b** are tag vectors corresponding to two different URLs. $\mathbf{a_i}$ and $\mathbf{b_i}$ are also frequency of i^{th} keywords in tag vectors of **a** and **b**.

K-means algorithm is used for clustering tag vectors using cosine similarity measure. We set **k** factor in k-means algorithm as the minimum of 7 and one tenth of search results size, as (2) shows.

$$k = \min(7, \left\lceil \frac{NumberOfSearchResults}{10} \right\rceil) \tag{2}$$

In the case of a full search result (100 pages) **k** would be set to 7. This is a simple intuitive formula which showed admissible results in the experiments. Finally, up to 7 clusters having more than 2 members are shown to the user, ordered by cluster size.

Cluster Labeling: After clusters are constructed, the cluster labels should be selected from cluster tags. For each cluster, tags with more frequency in the cluster and less frequency in other clusters are the best labels for describing the cluster. So we used TFIDF measure to select best labels for each cluster:

$$tfidf(w,c_i) = tf(w,c_i) * \log(\frac{N}{docFreq(w)}) \tag{3}$$

where *tf(w,ci)* is the number of tag vectors containing keyword **w** among tag vectors of cluster $\mathbf{c_i}$, **N** is the number of tag vectors (documents) and *docFreq(w)* is the total number of tag vectors containing keyword **w**. As you see, in (3) *tf(w,ci)* is not the sum of frequencies of **w** tag in $\mathbf{c_i}$ cluster, but the number of tag vectors in $\mathbf{c_i}$ containing keyword **w**. This is because if we use the frequencies, some tags that are used many times for a URL will dominate the cluster label.

For each cluster the four tags with most TFIDF are selected. These tags are candidates for participating in the cluster label. Each pair of these four tags is examined using Porter algorithm [19] and if they have the same root, the one with less occurrence is removed from the label. Finally a label with at least one keyword and at most four keywords is selected for each cluster.

After clusters are created and the labels are selected, the clusters are shown to the user. These are the top-level clusters and user can select one cluster to expand it. In this case the two phases of cluster construction and cluster labeling will be executed on documents in selected cluster, so the user can browse clusters hierarchically.

3 Experimental Results

We evaluated the proposed method by applying some different queries and comparing the results with the results of other SRC systems. The list of examined queries is created as a combination of some ambiguous queries (like java) and some clear queries (like "war of Iraq"). Some parts of this experimentation are shown in the following table along with results of Vivisimo:

Table 1. Created clusters for some queries

Query	Proposed Method	Vivisimo
Java	news articles daily blog	Developers Applications
	mac apple macosx macintosh	Java Technology
	book ebooks	Download
	linux gnu opensource freeware	Community
	html scripts webdesign webdev	Java Applet
	coffee shopping music	Implementation
	Simulation physics science animation	Language Programming
War of Iraq	government bush money oil	Cost
	casualties bush count humanrights	Bush
	2861 news bbc	Iran
	Fotografia gallery photo	Video, Attacks
	peace veterans activism global	Photos
Open Source	Cms php mambo content	Source Code
	applications linux oss community	Community
	Community business oss collaboration	Definition, Open Source
	apple osx macosx mac	Management
	html css xml xhtml	Distribution
	Geo geospatial gis maps	Open Source Database

In this table, clusters created by our method and those of Vivisimo are listed respectively for each query. Each row shows the label of one of the created clusters. Our method leads to seven top-level clusters for most of queries, but for some queries the number of created clusters is less than seven. Comparison of the proposed method with Vivisimo and other SRC services shows these differences:

- In many cases our method creates more distinctive clusters. For example consider "java" query: Vivisimo create vague clusters such as "Language Programming", "Java Technology", "Developers Applications" and "Implementation". These cluster labels remind similar concepts in the domain of java technology, so they are not distinctive labels.
- The proposed method is language independent. Most of similar systems, perform a language-based analysis of the snippets, e.g. they remove English stop words.

So these systems fail to construct meaningful clusters for non-English web pages. Although we also have a stemming stage in the preprocessing phase, this is only for the sake of tag refinement, so that to avoid similar tags in a cluster label.

- Our proposed method has better runtime performance because unlike other methods it doesn't use algorithms for text processing (stemming, stop word removal and etc) at runtime.

References

1. Zeng, H.-J., He, Q.-C., Chen, Z., Ma, W.-Y., Ma, J.: Learning to cluster web search results. In: Proc. of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, Sheffield, United Kingdom, July 25-29 (2004)
2. Golder, S., Huberman, B.A.: The Structure of Collaborative Tagging Systems. Technical report, In-formation Dynamics Lab, HP Labs (2005)
3. Brooks, C.H., Montanez, N.: Improved Annotation of the Blogosphere via Autotagging and Hierarchical Clustering. In: WWW 2006, Edinburgh, UK, May 23–26 (2006)
4. Osinski, S., Weiss, D.: A concept-driven algorithm for clustering search results. IEEE Intelligent Systems 20(3), 48–54 (2005)
5. Li, R., Bao, S., Yu, Y., Fei, B., Su, Z.: Towards effective browsing of large scale social annotations. In: WWW 2007, Banff, Alberta, Canada, May 8-12 (2007)
6. Wu, X., Zhang, L., Yu, Y.: Exploring social annotations for the semantic web. In: Proc. of the 15th international conference on World Wide Web, pp. 417–426 (2006)
7. Bao, S., Xue, G., Wu, X., Yu, Y., Fei, B., Su, Z.: Optimizing web search using social annotations. In: Proc. of the 16th international conference on World Wide Web, pp. 501–510. ACM Press, New York (2007)
8. Begelman, G., Keller, P., Smadja, F.: Automated Tag Clustering: Improving search and exploration in the tag space. In: WWW 2006, Edinburgh, UK, May 22–26 (2006)
9. Delicious, `http://del.icio.us`
10. Yahoo My Web, `http://myweb.yahoo.com/`
11. Kules, B., Kustanowitz, J., Shneiderman, B.: Categorizing web search results into meaningful and stable categories using fast-feature techniques. In: Proc. of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries, Chapel Hill, NC, USA (2006)
12. Vivisimo, `http://www.vivisimo.com`
13. Carrot Search Results Clustering Engine, `http://demo.carrot-search.com/`
14. Technorati, `http://technorati.com/`
15. Kummamuru, K., Lotlikar, R., Roy, S., Singal, K., Krishnapuram, R.: A hierarchical monothetic document clustering algorithm for summarization and browsing search results. In: Proc. of the 13th international conference on World Wide Web, pp. 658–665. ACM Press, New York (2004)
16. Ask Search Engine, `http://www.ask.com/`
17. MSN Live Search Engine, `http://www.live.com`
18. Open Directory Project, `http://www.dmoz.org/`
19. Porter, M.: An algorithm for suffix stripping. Program 14(3), 130–137 (1980)
20. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: Proc. of the Fifth Symposium on Math, Statistics, and Probability, pp. 281–297. University of California Press, Berkeley (1967)
21. Diigo social annotation service, `http://www.diigo.com/`

Event Detection from News Articles

Hassan Sayyadi, Alireza Sahraei, and Hassan Abolhassani

Department of Computer Engineering,
Sharif University of Technology, Tehran, Iran
{sayyadi,sahraii,abolhassani}@ce.sharif.edu

Abstract. In this paper, we propose a new method for automatic news event detection. An event is a specific happening in a particular time and place. We propose a new model in this paper to detect news events using a label based clustering approach. The model takes advantage of the fact that news events are news clusters with high internal similarity whose articles are about an event in a specific time and place. Since news articles about a particular event may appear in several consecutive days, we developed this model to be able to distinguish such events and merge the corresponding news articles. Although event detection is propounded as a stand alone news mining task, it has also applications in news articles ranking services.

Keywords: News Mining; Event Detection; Clustering; News Information retrieval; News Ranking.

1 Introduction

"Online news" is a special kind of common information existing on the web with its own characteristics. Characteristics that impose differences on some part of collection, search and mining process of traditional web search engines. These characteristics include: numerous reliable generating sources (high trust), daily updates, date of issue, etc. In this paper we investigate event detection service of news engines and a new efficient model and algorithm is proposed for this purpose. At first, we provide a comprehensive survey on news mining tasks. Afterwards, a new label-based model for news event detection is proposed.

Retrieval and ranking of news are important processes in news mining so that most of the search engines use traditional methods such as TF-IDF and vector similarity for these processes. However, they usually make some changes to adapt with news characteristics such as having time and title in consideration. In fact importance of a news article or in other words its hotness has also an essential role in news ranking [1]. There are many commercial news engines available but few academic researches have been done in this area. In spite of this lack of researches, suitable metrics are propounded; one of these metrics is the news article publication time. The fact is that the newer article is more important. On the other hand the importance of a news article is also depends on the number of relevant articles. The more number of news articles in a specific topic denote that the topic is more important. For this reason news articles are usually clustered in order to determine the cluster size and use this metric for ranking [2].

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 981–984, 2008.

In news mining literature one important assumption is that a hot news article is one with many relevant articles. But in this article we introduce the size of news event instead of number of relevant news. For instance there are plenty of news articles about Iraq but this fact does not imply that every article about Iraq is a hot news article. Similar articles are not necessarily in the same event; so for instance a news article about bombing in a small city in another country may be similar to bombing news in Iraq and this fact results in high rank assigning to this article because it has a large number of similar news articles. For these reasons we recommend that news events should be determined first and afterwards we could give higher ranks to events with more number of news. If a news article belongs to a more important event it will get a higher rank also.

Section 2 is a survey on related works. It follows by section 3 which introduces a general picture of our model. The label generation part of the model is discussed in section 4. Also in section 5 the event extraction part of the model is elaborated and Section 6 shows some experimental results.

2 Related Works

Although we use the event detection task as a means for news ranking, it has also other applications [3,4,5]. In [6,7] the most common approach of news event detection was proposed in which documents are processed by an on-line system. In these systems, when receiving a document, similarities between the incoming document and the already detected events (sometime represented by a centroid) are computed, and then a threshold is applied to determine whether the incoming document is the first story of a new event or an article of some known event.

Authors of [3] discover clusters at sentence level using hierarchical algorithms such as single-link, complete-link, and GroupWise-average. With assumption that news can point to different events, they presented that by clustering news in sentence level, each cluster points to an event. In the paper, wordnet[1] is employed to make more efficient comparison between words and sentences. It also uses one learning automata to consider the location of sentences in the document during clustering.

3 Proposed Model for Event Detection

In this paper we introduce a model for event detection from news articles. It is based on a label based clustering method. In fact we can treat news events as special kind of clusters with high internal similarity. Each of these clusters contains articles about an event in a particular time and place. Generally we extract events in news for each day and find their corresponding articles. However, there are events that have news articles spawning in several days; so we compare news events of consecutive days and merge them in the case of necessity. Our proposed method is based on this assumption:

[1] http://wordnet.princeton.edu/

- For each event in a day we can select several titles describing that event uniquely and every article related to that event contains at least one of those titles.

Considering this assumption the proposed algorithm includes the following major steps:

- Label generation
- Event extraction.

4 Label Generation

We introduce a label based event detection algorithm that generates events' labels in its first segment and use them to extract events in the second part. We extract noun phrases and score them and finally choose the appropriate ones for the event labels.

5 Event Extraction

In the second part of this method we extract news events based on the generated labels. To achieve this aim we first form the news article clusters. Afterwards the clusters are refined by using pruning and merging processes. In pruning step, we eliminate general clusters which do not point to a particular event. Up to the current step of algorithm several clusters may be generated for the same event. Therefore a merging process is necessary to achieve the final news events. We call this step, "same events merging". The currently presented steps are sufficient to extract news events in each day, but we need one more step to distinguish news events which last several days. In the final step we merge similar events of consecutive days.

6 Experimental Results

We implemented a retrieval engine for constructing a data set which is able to retrieve news articles from websites by RSS technology. We feed this engine with RSS address of five creditable news sites including: BBC, CNN, Associated Press, Reuters, and Agence France-Presse.

We store these data in MySql database. There are 5 news sources in this database and the average number of news articles in each day is about 1000.

For extracting noun phrases for events' noun labels we used the JMontylingua tool. As mentioned before we omit stop words and also terms with fewer than 3 letters. Fig. 1 shows that in average 1000 news article are retrieved from the 5 news sources in each day by using RSS technology and event detection process is performed on them. This figure shows that number of merged noun phrases is much less than total number of noun phrases. But the point is that more than 90% of detected events have noun labels consisting of several noun phrases. This result implies the importance of employing noun labels.

Since there is no available test dataset in the news mining field, we cannot compute precision and recall measures for our proposed algorithm. However in our observation for news articles in a period of 20 days, we found that the algorithm detected about 80% of news events correctly.

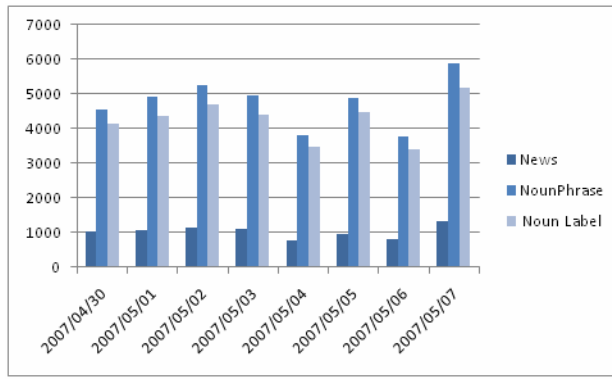


Fig. 1. Number of news articles, noun phrases, and labels

Acknowledgements. This research is partly supported by Iranian Telecommunication Research Center (ITRC).

References

1. Jinyi, Y., Jue, W., Zhiwei, L., Mingjing, L., Wei-Ying, M.: Ranking Web News Via Homepage Visual Layout and Cross-Site Voting. In: Lalmas, M., MacFarlane, A., Rüger, S.M., Tombros, A., Tsikrika, T., Yavlinsky, A. (eds.) ECIR 2006. LNCS, vol. 3936, pp. 131–142. Springer, Heidelberg (2006)
2. Corso, G.M.D., Gulli, A., Romani, F.: Ranking a Stream of News. In: 14th International Conference on World Wide Web, pp. 97–106. ACM Press, New York (2005)
3. Naughton, M., Kushmerick, N., Carthy, J.: Clustering Sentences for Discovering Events in News Articles. In: Lalmas, M., MacFarlane, A., Rüger, S.M., Tombros, A., Tsikrika, T., Yavlinsky, A. (eds.) ECIR 2006. LNCS, vol. 3936, pp. 535–538. Springer, Heidelberg (2006)
4. Atallah, M., Gwadera, R., Szpankowski, W.: Detection of Significant Sets of Episodes in Event Sequences. In: 4th IEEE International Conference on Data Mining, pp. 3–10. IEEE Computer Society, Washington (2004)
5. Li, Z., Wang, B., Li, M., Ma, W.Y.: A Probabilistic Model for Retrospective News Event Detection. In: 28th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 106–113. ACM Press, New York (2005)
6. Allan, J., Papka, R., Lavrenko, V.: On-line New Event Detection and Tracking. In: 21st annual international ACM SIGIR conference on Research and development in information retrieval, pp. 37–45. ACM Press, New York (1998)
7. Yang, Y., Pierce, T., Carbonell, J.G.: A Study on Retrospective and On-line Event Detection. In: 21st annual international ACM SIGIR conference on Research and development in information retrieval, pp. 28–36. ACM Press, New York (1998)

Architectural Styles as a Guide for Software Architecture Reconstruction

Kamyar Khodamoradi, Jafar Habibi, and Ali Kamandi

Computer Engineering Department, Sharif University of Technology,
Tehran, Iran
khodamoradi@ce.sharif.edu, jhabibi@sharif.edu,
kamandi@ce.shairf.edu

Abstract. Much effort has been put in defining methods, techniques, and tools for software architecture reconstruction (SAR)**,** Software Architecture Transformation (SAT), and Software Architecture-based Development, but much less attention has been paid to what lies at the heart of architecture-based development, *software architectural styles*. We argue that architecture-based software development could be much more productive if centered around the concept of software architecture styles, hence the need for style recovery in the process of software architecture reconstruction. The reason behind this is generally. Furthermore, with the coherence architectural styles can bring to the process of automated software development, integration of tools and methods might be accomplished more conveniently.

Keywords: Software Architectural Styles, Architecture-based Software Development, Automated Software Engineering.

1 Introduction

Software architecture reconstruction or recovery is a reverse engineering activity that aims at retrieving a documented architecture for an existing system [4]. This is in fact, the first of three essential steps of the horseshoe model of architecture-based software development, namely *Architecture Reconstruction/Recovery*, *Architecture Transformation*, and *Architecture-based Development*, outlined in [12][11]. For architecture reconstruction to be a successful activity, we need to identify the proper architecturally significant information and to extract it from the artifacts. The artifacts might include any document about the system architecture or behavior, but most of the times, the only reliable artifact available to work with, is the source code of the system.

We propose a process for architecture reconstruction that is based on software architectural styles. An architectural style bares an identification of software elements, namely component types, connectors that connect these components together, the overall topology of the software, the semantic rules by which elements are group or added to the system, and finally the activation model which describes the flow of control and data through the whole software system [1]. Software architecture styles play a role similar to that of design patterns, at a higher conceptual level [18][7]. For the

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 985–989, 2008.

purpose of incorporating these styles into our model, we suggest the use of a "style library" along side with other resources that contribute to the process of architecture reconstruction.

The reminder of this paper as organized as follows. First, we propose a model of software architecture recovery which is in fact, an extension of the horseshoe model. Then we outline the algorithm which lies at the heart of our suggested model. Section four, concludes the paper and also points out some of the potential directions that could be taken in upcoming researches on the subject.

2 Proposed Model

We propose to build a fifth layer on the top of architecture level of the Horseshoe model so as to further abstract the architecture to architectural styles. Our experiments show that incorporation of architectural styles into the process of architecture-based software development can make up for the lack of uniformity among several tools and technique utilized widespread across the business.

The fifth layer which we call *architectural style layer* requires another reengineering task by which a set of components and connectors together with their activation models and semantics are mapped into the form of architectural styles. The mapping must be carried in a way that it supports for the cases that a software system is composes of several sub-systems, each developed in a different architectural style.

A software architectural style transformation needs to be added for the fifth layer of abstract model. This kind of transformation occurs when the style of sub-system architecture does not meet its requirements and quality attributes associated with it.

The fifth layer practically serves the purpose of guiding the architecture recovery process from a high level more abstract point of view. The level of abstraction in turn is elevated through the transparency provided by the architectural style recognition algorithm discussed in the next section. The algorithm seeks to capture the most intrinsic features of the architecture represented with components, connectors, activation model, etc. and reflect these in features in independent modules of well-defined software architecture styles. As a result, the whole software is partitioned into a number of modules, for which the architectural styles are utterly understood. This last level of abstraction bears a couple of advantages:

1. The majority of current software products have been developed insight to software architectural styles. Since the architecture affects the whole software in so many ways, the proposed algorithm strives to reveal the basic characteristic of a system through architecture style recovery and make use of them in development phases.
2. Even if some software systems do not obey the rules of software architecture, it is still beneficial to look for patterns in the software that exhibit some sort of harmony in the design of the system. Since the most successful practices in software architecture are collected in the form of software architectural styles, the harmony in the design might imply the birth of new architectural styles.

For architecture style recovery, one can build a whole recovery process from scratch, including all file layers, or build a style recovery layer on top of the existing tools. Examples of existing tools include those built upon the CORUM II model presented in [12].

3 Algorithm

For an algorithm to be able to classify architectural styles of its input architectures, first we need to present the architecture someway. Then the presentation can be fed into an algorithm as an input parameter. But the representation of software architecture can be troublesome if it does not benefit from a general agreement among software engineering communities. The problem of integration of tools in reverse engineering has its roots in this kind of disagreement, therefore the first question to address, is to find a widely acceptable representation of software architecture.

Fortunately, it has already been done. As stated in [13], research community has come to a consensus on an Architecture Description Language (ADL), presented by a segment of the community of ACME as an architecture interchange language. Authors of [14] discuss that in order to meaningfully interchange architectural specifications across ADLs, a common basis for all ADLs must be established. This common basis is represented in form of a core including components, connectors, topology, ports and their roles, and re-maps which map a composite component or connector's internal architecture to elements of its external interfaces.

In ACME, any other aspect of architectural description is represented with property lists (*i.e.*, it is not core). With each component and connector is associated a *type*, *ports* (indicating its connectivity) and *semantics and constraints*. In proposed algorithm, we make use of the first two of these. Given a graph of the system with vertices representing components and the edges representing connectors, it sounds intuitively reasonable to check if the given graph matches the general topology of a known software architecture style. Unfortunately, this problem is known to be *NP-hard*. Dealing with topology matching problem, we will have to turn our attention to approximation algorithms for the problem. Besides, another issue needs to be handled. Software topology is not always decisive in determining the architectural style. For example, *blackboard* and *usual repository* styles share their topologies in common, but they are distinguished from each other mostly by their activation model. Therefore we use an algorithm based on component and connector types, and the only information about topology we incorporate into our algorithm, is their connectivity at their ports; to how many other components a single component is connected. The choice of using this knowledge is of course partly done for the sake of simplicity.

We presented a randomized algorithm based on component and connector types, and also simple connectivity constraints for a style topology. The style repository discussed earlier contains such information about the type and connectivity of components and connectors. The algorithm works as follows:

First, the algorithm strives to find a matching for the whole graph of the system at once. If the matching tests pass their thresholds, the system reports the corresponding style as the style for the whole system. Otherwise it tries to partition the system into two subsystems, finding styles for them at a time. In doing so, the algorithm first picks a random collection of components and connectors, and then tries to enhance it

by successive exchanges of the elements selected. The randomized element of the algorithm shows up right here. With some heuristics on how the algorithm should pick the elements in the first place, there can be good chance that it is not misled into an exponential-time futile search. The proposed algorithm is presented here:

Algorithm Style − Recovery

▷ ***input***: *an array of components* $comp_{arr}$,
an array of connectors $conn_{arr}$,
an array of styles $style_{arr}$,
$comp - type_{threshold}$,
$comp - port_{threshold}$,
$conn - type_{threshold}$,
$conn - port_{threshold}$.
▷ ***output***: *a list of subsystems and styles*, S

$S \leftarrow \emptyset$
for each *style* s ***in*** $style_{arr}$
do *match* $comp_{arr}.type$ ***and*** $s.comp - type$;
match $comp_{arr}.port$ ***and*** $s.comp - port$;
match $conn_{arr}.type$ ***and*** $s.conn - type$;
match $conn_{arr}.port$ ***and*** $s.conn - port$;
if *matching rates satisfy* $comp - type_{threshold}$, $comp - port_{threshold}$, $conn - type_{threshold}$, ***and*** $conn - port_{threshold}$
then $S \leftarrow S \cup s$; \\ meaning that the corresponding graph is also saved in some variable
else *partition the graph arbitraily into* 2 *subgraph* G_1 ***and*** G_2;
for each *style* s ***in*** $style_{arr}$ ***and*** G_1
do *swap the elements so that* G_1*matches* s
if G_1*matches* s
then $S \leftarrow S \cup s$;
repeat *the* ***else*** *part for reminder of graph*;
retuen S;

4 Conclusion and Future Work

The use of styles as a guide for software architecture reconstruction has proved to be remarkably helpful. Our experience showed satisfactory results for 3-tier traditional business applications. Yet, in order to tailor the approach to any conceivable architectural style, we need to go further. In particular, case studies need to be endeavored for finding better matching thresholds and more informed partitioning mechanisms. Also the impact of incorporating more detailed information about topology should be studied in forthcoming researches.

References

1. Albin, S.T.: The Art of Software Architecture: Design Methods and Techniques (2003)
2. Bengtsson, P., Bosch, J.: Scenario-based software architecture reengineering. In: Proceedings of the 5th International Conference on Software Reuse (ICSR5), 2–5 June 1998, pp. 308–317. IEEE Press, Los Alamitos (1998)
3. Bass, L., Clements, P., Kazman, R.: Software Architecture in Practice. Addison-Wesley, Reading (2004)

4. van Deursen, A., Hofmeister, C., Koschke, R., Moonen, L., Riva, C.: Viewpoints in Software Architecture Reconstruction. In: Proceedings. Fourth Working IEEE/IFIP Conference on Software Architecture (WICSA), pp. 122–132 (2004)
5. Finnigan, P.J., Holt, R., Kalas, I., Kerr, S., Kontogiannis, K., Mueller, H., Mylopoulos, J., Perelgut, S., Stanley, M., Wong, K.: The Portable Bookshelf. IBM Systems Journal 36(4), 564–593 (1997)
6. Feijs, L.M.G., van Ommering, R.C.: Relation Partition Algebra—Mathematical Aspects of Uses and Part-Of Relations. Science of Computer Programming 33, 163–212 (1999)
7. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software (1994)
8. Garlan, D., Monroe, B., Wile, D.: ACME: An interchange language for software architecture, 2nd edn., Technical Report CMU-CS-95-219, Carnegie Mellon University (1997)
9. Hassan, A.E., Holt, R.C.: Architecture Recovery of Web Applications. In: Proceedings of the 24th International Conference on Software Engineering, pp. 249–259 (2002)
10. Hoffmeister, C., Nord, R., Soni, D.: Applied Software Architecture. Addison-Wesley, Reading (2005)
11. Kazman, R., Burth, M.: Assessing Architectural Complexity. In: Proceedings of 2nd Euromicro Working Conference on Software Maintenance and Re-engineering (CSMR 1998), pp. 104–112 (1998)
12. Kazman, R., Woods, S.G., Carrière, S.J.: Requirements for Integrating Software Architecture and Reengineering Models: CORUM II. In: Proceedings of the Working Conference on Reverse Engineering (WCRE), p. 154 (1998)
13. Medvidovic, N.: A Classification and Comparison Framework for Software Architecture Description Languages, Technical Report UCI-ICS-97-02, University of California at Irvine (1997)
14. Mendonça, N.C., Kramer, J.: Architecture Recovery for Distributed Systems. In: SWARM Forum at the Eighth Working Conference on Reverse Engineering, Stuttgart, Germany, October 2-5, 2001. IEEE Computer Society, Los Alamitos (2001)
15. Shaw, M., Garlan, D.: An Introduction to Software Architecture (1994)
16. Stoermer, C., O'Brien, L., Verhoef, C.: Moving towards quality attribute driven software architecture reconstruction. In: Proceedings of the 10th Working Conference on Reverse Engineering (WCRE), Victoria, Canada, November 2003. IEEE Computer Society Press, Los Alamitos (2003)
17. Tahvildari, L., Kontogiannis, K., Mylopoulos, J.: Quality-driven software re-engineering. Journal of Systems and Software 6(3), 225–239 (2003)
18. Woods, S., Quilici, A., Yang, Q.: Constraint-Based Design Recovery for Software Reengineering: Theory and Experiments. Kluwer Academic Publishers, Dordrecht (1998)

Best Effort Flow Control in Network-on-Chip

Mohammad S. Talebi[1], Fahimeh Jafari[1,2], Ahmad Khonsari[1,3], and Mohammad H. Yaghmaee[2]

[1] IPM, School of Computer Science, Tehran, Iran
[2] Ferdowsi University of Mashhad, Mashahhad, Iran
[3] ECE Department, University of Tehran,Tehran, Iran
mstalebi@ipm.ir, jafari@ipm.ir, ak@ipm.ir, hyaghmae@ferdowsi.um.ac.ir

Abstract. Research community has recently witnessed the emergence of Interconnect networks methodology based on Network-on-Chip (NoC). Employing end-to-end congestion control is becoming more imminent in the design process of NoCs. This paper presents a centralized congestion scheme in the presence of both elastic and streaming flow traffic mixture. In this paper, we model the desired BE source rates as the solution to a utility maximization problem while preserving GS traffics services requirements at the desired level. We propose an iterative algorithm as the solution to the maximization problem which has the benefit of low complexity and fast convergence.

Keywords: Flow Control, Utility-based optimization, Gradient Method.

1 Introduction

Quality-of-Service (QoS) provisioning in NoC's environment has attracted many researchers and currently it is the focus of many literatures in NoC research community. NoCs are expected to serve as multimedia servers and are required not only to carry Elastic Flows, i.e. Best Effort (BE) traffic, but also Inelastic Flows, i.e. Guaranteed Service (GS) traffic which requires tight performance constraints such as necessary bandwidth and maximum delay boundaries [1].

Flow control is still a novel problem in NoCs and to the best of our knowledge only few works has been carried out in this field. So far, several works have focused on this issue for NoC systems. In [2], a prediction-based flow-control strategy for on-chip networks is proposed in which each router predicts the buffer occupancy to sense congestion. This scheme controls the packet injection rate and regulates the number of packets in the network. In [3] link utilization is used as a congestion measure and a Model Prediction-Based Controller (MPC), determines the source rates. Dyad [4] controls the congestion by using adaptive routing when the NoC faces congestion.

In this paper, we focus on the flow control for Best Effort traffic as the solution to a utility-based optimization problem. To the best of our knowledge, none of the aforementioned works have dealt with the flow control problem through utility optimization approach.

This paper is organized as follows. In Section 2 we present the system model and formulate the underlying optimization problem for BE flow control. In section 3 we

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 990–993, 2008.

propose the optimal flow control algorithm. Section 4 presents the simulation results. Finally, section 5 concludes the paper and states some future work directions.

2 System Model and Flow Control Problem

We model the flow control in NoC as the solution to an optimization problem. As in [5], we consider NoC as a network with a set of bidirectional links L and a set of sources S. A source consists of Processing Elements (PEs), routers and Input/Output ports. Each link $l \in L$ is a set of wires, busses and channels that are responsible for connecting different parts of the NoC and has a fixed capacity of c_l packets/sec. We denote the set of sources that share link l by $S(l)$.

We denote the set of sources with BE and GS traffic by S_{BE} and S_{GS}, respectively. Each link l is shared between the two aforementioned traffics. GS sources will obtain the required amount of the capacity of links and BE sources benefit from the remainder. Our objective is to choose source rates with Best Effort traffic so that to maximize the sum of the logarithm of the BE source rates as following [5]:

$$\max_{x_s} \sum_{s \in S_{BE}} \log x_s \tag{1}$$

subject to:

$$\sum_{s \in S_{BE}(l)} x_s + \sum_{s \in S_{GS}(l)} x_s \le c_l \qquad \forall l \in L \tag{2}$$

$$x_s > 0 \qquad \forall s \in S_{BE} \tag{3}$$

Problem (1) is a convex optimization problem and therefore admits a unique maximizer [6]. Due to coupled nature of problem (1), it should be solved using centralized methods like Interior Point method which poses great computational overhead onto the system [6] and hence is of little interest. One way to reduce the computational complexity is to transform the constrained optimization problem into its *Dual*, which can be defined to be unconstrained. According to the Duality Theory [6], each convex optimization problem has a dual, whose optimal solution, leads to the optimal solution of the main problem.

3 Optimal Flow Control Algorithm

In order to solve problem (1), we have obtained its Dual function through which the Dual problem can be achieved. As the Dual problem is unconstrained, it can be solved using simple iterative methods such as Gradient Projection Method [6]. We have solved the Dual problem using Projected Gradient Method, in an iterative manner. Due to lack of enough space we have omitted mathematical derivations. We direct serious readers to the extended version of this work [7]. The iterative solution of the Dual problem can be used as an iterative flow control algorithm for BE traffic, which is listed below as algorithm 1.

Algorithm 1: Flow Control for BE Traffics in NoC

Initialization:

1. Initialize c_l of all links.
2. Set link shadow price vector to zero.
3. Set the ε as the stopping criteria.

Loop:

Do until $(\max|x_s(t+1)-x_s(t)|<\varepsilon)$

1. $\forall l \in L$: Compute new link prices:

$$\lambda_l(t+1)=\left[\lambda_l(t)-\gamma\left(c_l(t)-\sum_{s\in S_{GS}(l)}x_s(t)-\sum_{s\in S_{BE}(l)}x_s(\lambda(t))\right)\right]^+$$

2. Compute new BE source rates as follows

$$x_s(t+1)=\frac{a_s}{\sum_{l\in L(s)}\lambda_l(t+1)}$$

Output:

Communicate BE source rates to the corresponding nodes.

4 Simulation Results

In this section we examine the proposed flow control algorithm, listed above as Algorithm 1, for a typical NoC architecture. In our scenario, we have used a NoC with 4×4 Mesh topology which consists of 16 nodes communicating using 24 shared bidirectional links; each one has a fixed capacity of 1 Gbps. In our scheme, packets traverse the network on a shortest path using a deadlock free XY routing.

As a performance metric, the relative error with respect to optimal source rates for different step sizes is shown in Fig. 1. The relative error is averaged over all active

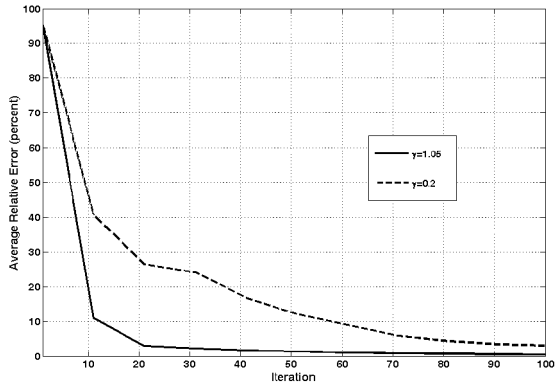


Fig. 1. Average Relative Error for $\gamma=1.05$ and $\gamma=0.2$

sources. This figure reveals that for $\gamma = 1.05$ the algorithm leads to less than 10% error in average just after about 13 iteration steps, and after 20 steps the average error lies below 5%. However, for a smaller step size, e.g. $\gamma = 0.2$, these thresholds are about 60 and 75 steps, respectively.

5 Conclusion and Future Works

In this paper we addressed the problem of flow control for BE traffic in NoC systems. Flow control was considered as the solution to a utility maximization problem which was solved indirectly through its dual using gradient projection method. This was led to an iterative algorithm that can be used to determine optimal BE source rates. The algorithm can be implemented by a controller which poses a light communication and communication overhead to the system. Further investigation about the effect of delay incurred by the proposed algorithm and convergence behavior are the main directions of our future studies.

References

1. Benini, L., DeMicheli, G.: Networks on Chips: A New SoC Paradigm. Computer 35(1), 70–78 (2002)
2. Ogras, U.Y., Marculescu, R.: Prediction-based flow control for network-on-chip traffic. In: Proceedings of the Design Automation Conference (2006)
3. Van den Brand, J.W., Ciordas, C., Goossens, J.W., Basten, T.: Congestion-Controlled Best-Effort Communication for Networks-on-Chip. In: Proceedings of Design, Automation and Test in Europe Conference, pp. 948–953 (2007)
4. Jingcao, H., Marculescu, R.: DyAD - smart routing for networks-on-chip. In: Design Automation Conference, pp. 260–263 (2004)
5. Low, S.H., Lapsley, D.E.: Optimization Flow Control, I: Basic Algorithm and Convergence. IEEE/ACM Transactions on Networking 7(6), 861–874 (1999)
6. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2004)
7. Talebi, M.S., Jafari, F., Khonsari, A., Yaghmaee, M.H.: Weighted Proportionally Fair Flow Control for Best Effort Traffic in Network–On-Chip. In: Proceedings of IPDPS (to appear, 2007)

Prevention of Tunneling Attack in endairA

Mohammad Fanaei, Ali Fanian, and Mehdi Berenjkoub

Department of Electrical and Computer Engineering,
Isfahan University of Technology (IUT), Isfahan, Iran
{m_fanaei,fanian}@ec.iut.ac.ir, brnjkb@cc.iut.ac.ir

Abstract. endairA is one of the most secure on-demand ad hoc network source routing protocols which provides several defense mechanisms against so many types of attacks. In this paper, we prove the vulnerability of endairA to the tunneling attack by presenting an attack scenario against it. We also propose a new security mechanism to defend it against the tunneling attack by the utilization of the delay between receiving and sending its control packets computed locally by all intermediate nodes. Our proposed security mechanism can detect probable tunnels in the route as well as approximate locations of the adversarial nodes. It needs no time synchronization between mobile nodes of the network. It also does not change the number of control packets involved in endairA and only modifies the RREP messages slightly.

Keywords: Ad hoc network, secure routing, endairA, tunneling attack, delay.

1 Introduction

endairA [1] is a secure on-demand source routing protocol based on *Dynamic Source Routing* (DSR) [2]. It utilizes digital signatures to authenticate intermediate nodes in the unicast process of route replies. Although it provides defense mechanisms against so many types of attacks, it is vulnerable to the tunneling attack. Tunneling means that two, potentially remote, adversarial nodes pass control packets, including *Route Request* (RREQ) and *Route Reply* (RREP), between each other by encapsulating them into normal data packets and using the multi-hop routing service offered by the network to transfer those data packets [3]. The result of the tunneling attack is that two adversarial nodes at the tunnel ends appear to be neighbors. Therefore, the routes through the tunnel may be selected by some victim nodes as the optimum ones and this will increase adversarial control over the communications of those nodes.

In Section 2 of this paper, the vulnerability of endairA to the tunneling attack is proved by presenting an attack scenario in an ad hoc network. Section 3 presents our proposed security mechanism to defend endairA against the tunneling attack by exploiting the delays measured by intermediate nodes between receiving the RREQ and receiving the corresponding RREP. At the end of this section, the results of our detailed security analysis of the proposed tunneling prevention mechanism will be provided to prove its applicability in ad hoc networks. Finally in Section 4, the paper and its contributions are concluded.

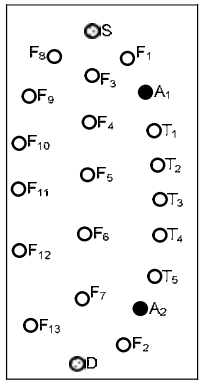
H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 994–999, 2008.

2 Vulnerability of endairA to the Tunneling Attack

To prove the vulnerability of endairA to the tunneling attack, let us consider the network configuration illustrated in Fig. 1. Assume that the source node S initiates a route discovery process to find a valid route toward the destination node D utilizing endairA. If the network was free of any adversarial node, the route $SF_3F_4F_5F_6F_7D$ would be selected by the source as the optimum route between it and the destination because it is the shortest route between them. Nevertheless, in the presence of two adversarial nodes with the compromised identifiers A_1 and A_2, the route between the source and the destination is diverted and passes through them by launching the tunneling attack against the routing protocol, endairA. The tunneling attack scenario against endairA is summarized in Fig. 1 in which *id* is a randomly generated request identifier and sig_X is the signature of the node *X* on all the preceding fields of the packet. As it is apparent from this Figure, when the RREQ reaches the first adversarial node A_1, it places the RREQ in the payload, *M*, of a normal data packet, *DATA*, and forwards it through the multi-hop route *RT* toward the other adversarial node A_2. Having received the tunneled RREQ, A_2 extracts it from the payload of the data packet and continues the normal operations of endairA. With the same procedure, the RREP will be tunneled by A_2 in a normal data packet, *IM*, through the multi-hop route *IRT* toward A_1.

According to the specifications of endairA [1], when an intermediate node (or the source node) receives the RREP, it determines if its identifier is in the node list carried by the RREP. Then it verifies that the preceding node (or the source if it is the first node in the node list) and the following node (or the destination if it is the last node in the node list) whose identifiers are before and after that of the node, are its neighbors. (The source verifies only its neighborhood with the first node in the node list.) After that, the node verifies the signatures of the following nodes in the node list and that of the destination which are included in the RREP. It also verifies that all of the signatures correspond to the following identifiers included in the node list and to the destination's. Failing only one of these verifications is sufficient for the node to drop the RREP. Nevertheless, if all these verifications are successful, the intermediate node will then append its signature computed over the whole RREP (including previous signatures) and forwards it to the preceding node in the node list or to the source or if it is the source node, it will accept obtained route from the received RREP as a valid route toward the destination and will cache it. In the future, the source will use this route to forward data packets toward that destination.

In our attack scenario, since all of the above mentioned verifications are successful in all intermediate nods outside of the tunnel and especially in the source node, it accepts the route $F_1A_1A_2F_2$ obtained from the node list carried by the returned RREP as a valid route toward the destination. However, this route passes through two adversarial nodes launching the tunneling attack against the routing protocol and is not a valid route in the network topology.



S → *: (RREQ, S, D, id, [ ])
F_1 → *: (RREQ, S, D, id, [F_1])
A_1 → T_i: (DATA, RT, M) , RT= $A_1T_1T_2T_3T_4T_5A_2$
M=(RREQ, S, D, id, [F_1, A_1])
T_i → T_{i+1}: (DATA, RT, M) , for i=1, 2, 3, 4
T_5 → A_2: (DATA, RT, M)
A_2 → *: (RREQ, S, D, id, [F_1, A_1, A_2])
F_2 → *: (RREQ, S, D, id, [F_1, A_1, A_2, F_2])
D → F_2: (RREP, S, D, id, [F_1, A_1, A_2, F_2], [sig_D])
F_2 → A_2: (RREP, S, D, id, [F_1, A_1, A_2, F_2] , [sig_D, sig_{F2}])
A_2 → T_5: (DATA, IRT, IM) , IRT=$A_2T_5T_4T_3T_2T_1A_1$
IM=(RREP, S, D, id, [F_1, A_1, A_2, F_2] , [sig_D, sig_{F2}, sig_{A2}])
T_{i+1} → T_i: (DATA, IRT, IM) , for i=1, 2, 3, 4
T_1 → A_1: (DATA, IRT, IM)
A_1 → F_1: (RREP, S, D, id, [F_1, A_1, A_2, F_2], [sig_D, sig_{F2}, sig_{A2}, sig_{A1}])
F_1 → S: (RREP, S, D, id, [F_1, A_1, A_2, F_2], [sig_D, sig_{F2}, sig_{A2}, sig_{A1}, sig_{F1}])

Fig. 1. (a) Network configuration where two adversarial nodes A_1 and A_2 launch the tunneling attack against endairA. (b) Tunneling attack scenario against andairA. * indicates that the message has been broadcasted to all the node's neighbors.

3 Proposed Security Mechanism to Prevent Tunneling Attack

In this section, the messages involved in endairA are slightly modified. Some timing information is added to the RREP messages to make endairA resistant against the tunneling attack. This timing information consists of the delay between receiving the *last* bit of the RREQ and receiving the *first* bit of the corresponding RREP as well as the delay between receiving the last bit of the RREQ and rebroadcasting its *first* bit, both of them computed by all intermediate nodes. With regard to the fact that the difference between the delays measured by two successive intermediate nodes must not exceed a well-defined threshold, the source will later use these delays to detect any probable tunnel in the route obtained from the RREP as well as the approximate location of the adversarial nodes constructing such tunnels. The operation of the proposed approach is as follows:

Each intermediate node, such as I, saves the time in which it receives the *last* bit of the RREQ for the first time, $T_{I,QR}$, the time in which it rebroadcasts the *first* bit of the received RREQ, $T_{I,QB}$, as well as the time in which it receives the *first* bit of the corresponding RREP, $T_{I,PR}$, in its routing table. It must be noted that the time in which I succeeds to rebroadcast its received RREQ depends on its experienced channel conditions and specifically on congestion in its links in the network and is determined by IEEE 802.11 [4].

I subtracts $T_{I,QR}$ from $T_{I,QB}$ to compute the *one-hop-delay* between receiving the last bit of the RREQ and rebroadcasting its first bit, OHD_I, which is also saved in its routing table. By subtraction of $T_{I,QR}$ from $T_{I,PR}$, I computes the delay between receiving the last bit of the RREQ and receiving the first bit of the corresponding RREP, D_I. It must be noted that for the destination node, this delay is defined as the delay between receiving the last bit of the RREQ and sending the first bit of the corresponding

RREP, D_D. Furthermore, by such definitions for $T_{I,QR}$, $T_{I,QB}$ and $T_{I,PR}$, the delays OHD_I and D_I do not include the transmission time in which the RREQ or the RREP is being transmitted toward I.

In the next step, I appends D_I and OHD_I to the new field added to the RREP messages involved in endairA. The remaining fields of the RREP as well as the remaining operations performed by the intermediate nodes are exactly the same as those in the original version of endairA [1]. The basic operations and message format of endairA with the proposed defense mechanism against the tunneling attack have been illustrated in Fig. 2.

Receiving the RREP, the source first verifies all the verifications mandated by the specifications of endairA [1]. Then, it considers the delays included in the received RREP. These delays will be analyzed by the source to detect any probable tunnel in the route as well as to determine the approximate location of the tunnel ends. To decide whether the route obtained from the RREP is acceptable or it is a tunneled one, the source must verify the following criteria:

R_1: For every X in the node list, $OHD_X \leq \Delta$ where Δ is the maximum acceptable delay in the MAC layer during which the sender node must send its received control packet toward its intended neighbor. If the forwarding delay exceeds Δ and the packet does not reach its intended destination during this time, it will be assumed that the intended destination is unreachable. This criterion is a straightforward rule which states that the one-hop-delay of any intermediate node must be less than the maximum acceptable delay during which the intermediate node X must succeed to rebroadcast its received RREQ. If it does not succeed to do so, it will drop the RREQ.

R_2: For every X and Y in the node list that X is closer to the source than Y, then $D_X > D_Y$. This criterion is another straightforward rule which states that the closer the node is to the source, the greater delay it experiences between receiving the last bit of the RREQ and receiving the first bit of the corresponding RREP.

R_3: For every X and Y in the node list which are neighbors and X is closer to the source than Y, then $D_X \leq D_Y + OHD_X + \Delta$. This criterion specifies the range in which the difference between the delays of two successive nodes in the node list must fall. More precisely, the delay experienced by X between receiving the RREQ and receiving the corresponding RREP can exceed that of Y by two terms. The first one is the delay which it experiences before it succeeds to rebroadcast its received RREQ, OHD_X. The second one is the delay experienced by Y before it succeeds to send the corresponding RREP toward X. According to IEEE 802.11, this delay must be less than Δ. Therefore, the inequality of $D_X \leq D_Y + OHD_X + \Delta$ must be satisfied for D_X and D_Y.

To disrupt the functionality of our proposed tunneling prevention mechanism, the adversarial nodes at the tunnel ends tend to maliciously report their measured delays falsified. However, since they must obey the above mentioned criteria, they cannot report any preferred value for their reported delays. Our detailed security analysis of the proposed tunneling prevention mechanism showed that, in the worst case, if A_2 reports Δ as its one-hop-delay and $D_{F2}+2\Delta$ as its delay between receiving the last bit of the RREQ and receiving the first bit of the RREP, and also A_1 does the same and reports Δ as its one-hop-delay and $D_{A2}+2\Delta$ as its delay between receiving the last bit

of the RREQ and receiving the first bit of the RREP, the tunnel will be detected by the source provided that the delay of passing control packets through the tunnel between A_1 and A_2 in the RREQ broadcasting phase as well as the RREP returning back toward the source exceeds 5Δ.

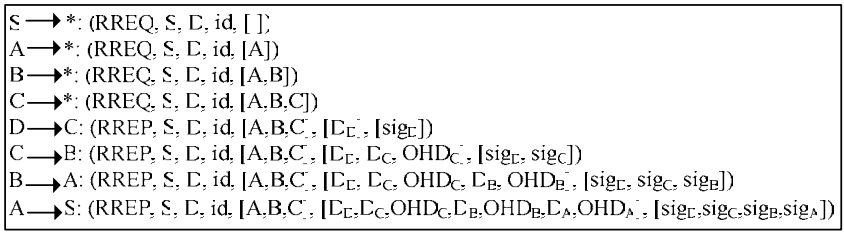


Fig. 2. Basic operations and message format of endairA with the proposed defense mechanism against the tunneling attack. Each signature in the messages is computed over all the preceding fields (including other signatures).

Although our tunneling prevention mechanism fails to detect *short* tunnels, it will limit the adversary's ability to launch undetected tunneling attacks to an acceptable level. With regard to the fact that tunneling has a meaningful impact on routing functionality only if the routes through the tunnels are much longer than those outside of them, failing to detect short tunnels is not a major drawback for our proposal.

As you may have noticed, when the source detects the tunnel in the route obtained from the RREP, it can determine the approximate location of the adversarial nodes launching tunneling attack in a one hop neighborhood.

The proposed approach does not need any time synchronization between the mobile nodes of ad hoc network. In other words, both of the delays OHD_I and D_I computed by an intermediate node are independent of the real time in other nodes and this operation is performed locally. Furthermore, it is a cross-layer approach in which the MAC layer information has been exploited in the network layer operation and signaling.

4 Conclusions

In this paper, the vulnerability of endairA to the tunneling attack has been proved. Then our proposed security mechanism has been presented to prevent this type of attack. The proposed approach exploits the delay between receiving the RREQ and receiving the corresponding RREP as well as the one-hop-delay between receiving the RREQ and rebroadcasting it measured by all intermediate nodes. Since these delays are computed locally by the nodes, our proposal does not need any time synchronization between mobile nodes of the network. It also does not change the number of control packets involved in endairA and only modifies the RREP messages slightly. Our detailed security analysis of the proposed approach shows that it will drastically decrease the possibility of the tunneling attack against endairA.

References

1. Acs, G., Buttyan, L., Vajda, I.: Provably Secure On-demand Source Routing in Mobile Ad Hoc Networks. IEEE Trans. on Mobile Computing 5(11), 1533–1546 (2006)
2. Johnson, D.B.: Routing in Ad Hoc Networks of Mobile Hosts. In: Proc. of the IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 1994), pp. 158–163 (1994)
3. Buttyan, L., Hubaux, J.P.: Security and Cooperation in Wireless Networks, 1st edn. Cambridge University Press, Cambridge (2007)
4. IEEE 802.11 Working Group: WLAN Medium Access Control and Physical Layer Specifications. Standard Specifications (1999)

Digital Social Network Mining for Topic Discovery

Pooya Moradianzadeh, Maryam Mohi, and Mohsen Sadighi Moshkenani

Dept. of IT, Sharif University of Technology – International Campus, Kish, Iran
moradianzadeh@kish.sharif.edu, mohi@kish.sharif.edu,
sadighim@cc.iut.ac.ir

Abstract. Networked computers are expanding more and more around the world, and digital social networks becoming of great importance for many people's work and leisure. This paper mainly focused on discovering the topic of exchanging information in digital social network. In brief, our method is to use a hierarchical dictionary of related topics and words that mapped to a graph. Then, with comparing the extracted keywords from the context of social network with graph nodes, probability of relation between context and desired topics will be computed. This model can be used in many applications such as advertising, viral marketing and high-risk group detection.

Keywords: Web Mining, Social Network, Data Mining, Topic identification.

1 Introduction

Nowadays, internet and web services rapidly grow and in parallel, digital social networks get an important role in people real life [1]. In fact, digital social networks are interactive networks that use internet as a media for making a relation between human. Emails, Weblogs and IM are instances of social networks. With rapidly growing users of this networks, mining in the large-scale of data, can provide better and efficient usage of hidden potential of this type of networks [1]. Results of that can be used in different applications such as marketing [1, 2] and advertising [3]. In this paper, we are going to extract and classify keywords from content of data exchanges by users, to discover the topics that social network users transmitted.

In the next section, related work in mining of social networks will be reviewed. In section 2, our model would be discussed and in section 3, we will evaluate our model.

1.1 Related Work

In recent years, social network as computer-mediated system that make interrelations between people attract researchers [1]. The researches in this area include considerable commercial applications such as viral marketing [2], advertise management [3], community discovery [4] so Email and IM mining [1, 5, 6]. In the mentioned applications, the main target of analysis is finding a group of people that have similar properties, such as similar events or friends [1, 2]. Generally, the basic method of analyzing social networks is to map the activity of networks to a graph as G(V,E) that V is the set of entities in the

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 1000–1003, 2008.

network and E is the set of links between these entities [1]. There are many different models for creating that graph, but some properties are useful for computing the intensity of relation between entities in all models like Connectedness, Betweenness and Closeness [7]. We are going to discover the topic of exchanged information in digital social network; this target can lead us to finding groups that transmit similar topics.

2 Proposed Model

To determine the level of the dependency between each word and desired topics, relation between them must be defined. Therefore, we define a hierarchical dictionary of the words and related topics that can be divided into subtopics, linking those together makes a directional graph while the height of graph is at most 4.

$$\begin{cases} T=\{T_{1,0},T_{2,0},\ldots,T_{n'',0}\}=\text{ set of topics} \\ W=\{w_1,w_2,\ldots,w_m\}=\text{set of words} \\ T_{i,0}=\{T_{1,i,i,1},T_{2,i,i,1},\ldots,T_{n',i,i,1}\} \\ T_{i,j,k,l}=\{T_{1,i,k,l+1},T_{2,i,k,l+1},\ldots,T_{n,i,k,l+1}\} \\ \forall t\in T_{i,j,k,l}\nexists t\in T'_{i,j,k,l}, if\ \ i\neq j\forall w_i,w_j\in W\nexists w_i=w_j \end{cases}, \begin{cases} i: ID\ of\ \ topic\ in\ the\ subset \\ j: ID\ of\ \ top\ set \\ k: ID\ of\ \ main\ set \\ l: the\ level\ of\ \ subset,\ 0\le l\le 3 \\ (i,j,k,l,n,n',n''\in \square) \end{cases} \tag{1}$$

It is expected that some words found, does not exist in this dictionary. Hence, we define Dynamic set as a set of unknown words. This set consists of the Keyword set, as set of extracted keywords from context, and probabilities of relation between topics and keywords. Another important issue is general words, which does not show any special topic. Calculating the dependency level for them is useless then with adding these words to a Filtering set, we remove it from Keyword set.

$$\begin{aligned} &D=\ set\ of\ \ unknown\ concept=\{D_1,D_2,\ldots,D_n\} \\ &K=\{extracted\ \ keywords\ from\ exchanges\ context\}=\{z_1,\ldots z_i,z_j\},z_i\in W,z_j\notin W \\ &D_c=\{(z_j,K-\{z_j\}),(T_{i,j,k,l},n),(T_{i',j,k,l},n'),\ldots\},n=Prob.\ of\ \ relation\ between\ (K\ \&\ T) \end{aligned} \tag{2}$$

In next step, weight must be assigned to the edges. This weight is a value in [0,1] and computed based on two main measures:

- Link stability: number of paths between source and destination
- Distance between nodes: number of nodes in each paths

Calculating weight of links between words and topics lead to computing the probability of relation between context and topic. Our formula shows that with increasing the link stability, the weight value increases and with increasing the distance between nodes in each path, the rate of increases decrease.

$$\begin{cases} \text{d= distance between } V_{Source}\ \&V_{Destination}\ \text{ in each path and } 0\le d\le 3 \\ \text{n = number of paths between } V_S\ \&V_D\ \text{ with the same distance} \\ e=n\times(d+1) \\ f(d,e)=r_d(V_S,V_D)=\left(1-d/e\right) \end{cases} \tag{3}$$

$$\begin{cases} R(V_S,V_D)=\{r_i(V_S,V_D),r_j(V_S,V_D),r_k(V_S,V_D)\} \\ r_i(V_S,V_D)\geq r_j(V_S,V_D)\geq r_k(V_S,V_D),1\leq i,j,k\leq 3,i\neq j\neq k \\ r_1=r_i(V_S,V_D),r_2=r_j(V_S,V_D),r_3=r_k(V_S,V_D) \\ f(R(V_S,V_D))=f(r_1,r_2,r_3)=r_1+(1-r_1)\cdot(r_2+(1-r_2)\cdot(r_3)),0\leq f(R(V_S,V_D))\leq 1 \\ R=\{R(V_i,V_D)\},V_i\in K,V_D\in T_{i,j,k,l} \\ P(V_D)=\dfrac{\sum_{i=1}^{N} f(R(V_i,V_D))}{N},\ N=\text{number of entities in keyword set} \end{cases} \quad (4)$$

According to past sections, the algorithm presented for our proposed model can be explained in the following steps:

1.Create dictionary according to (1)

2.Create Keyword set from extracted context as defined in (2)

3.Removing General words

4.***if*** $\exists\ V_i \in G(V,E)$ for $z_j \in K$ ***then*** Calculate weight and store it in R else

$\{$***if*** $\exists\ V_i \in D$ ***then if*** $\{(z_j \in D_i, D_i \subseteq D)$ ***and*** $(K_{D_i} \in D_i, K_{D_i} \subseteq K_i)$ ***then Weight*** add to R

and $(D_j$ in $K_i)$ exchange with $z_j\}$ ***else*** $\{k_i = k_i - z_j$ ***and*** z_j attached to D as defination (2)$\}\}$

5.Calculate the probability of relation for each topic

if $z_j \notin k_i$ ***then*** attach this probability to D as definition (2).

3 Evaluation

For evaluating this model, first a dictionary of data is created based on the definitions. This dictionary, consist of 25 root topics such as Computer, Internet and Commerce. Each of them was divided into some subtopics. Words collected from glossary of each topic, so for some topics such as Mobile, different brands collected from websites. Finally, the set of words would be contained with 5200 words and Filtering set with 100 words. Test data gathered from our email archive from 2002 to 2007. We developed a program with Delphi language that removes the extra information from emails and compute the P(D) according to our model.

After finishing this process, 400 unknown words found on Dynamic set, analysis on this set shows that, most of these words have wrong spelling and some of these words are names of persons. Figure 1, shows the distribution of the three top ranked topics in 100 random contexts. We analyzed these 100 contexts manually, for checking the confidence level of this model, in 88 contexts, ranked topics were correct. We predict that if problem of wrong spellings were solved then the rate of success of this model would be increased.

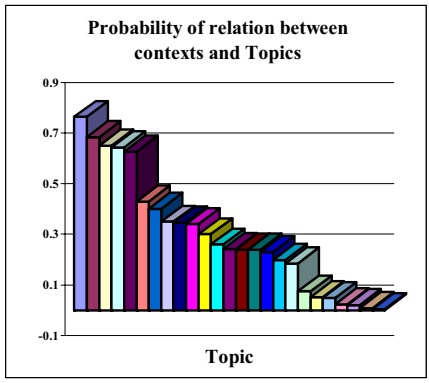


Fig. 1. Result of Mining shows the 25 topic and probability of relation between context

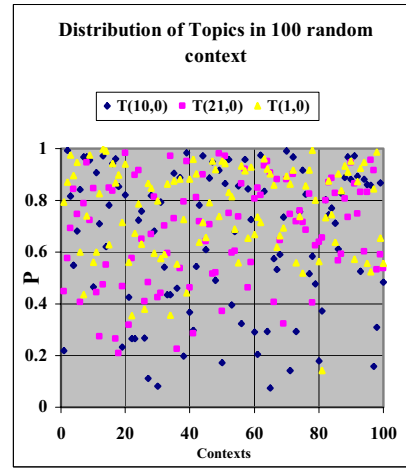


Fig. 2. Distribution of 3 top ranked topic

4 Conclusion

In this paper, we propose a model for finding the topics that social network users interact with their messages. This method could be used in different applications specially for classifying the users of social network. This classification is very useful for advertising and marketing. A result of testing shows that this model could discover the topic from context with high confidence level (about 88%). In future, we are going to improve this model for developing the semantic advertisement systems.

References

1. Staab, S., Domingos, P., Mike, P., Golbeck, J., Ding, L., Finin, T.T., Joshi, A., Nowak, A., Vallacher, R.: Social networks applied. IEEE Intelligent Systems 20, 80–93 (2005)
2. Richardson, M., Domingos, P.: Mining knowledge-sharing sites for viral marketing. In: 8 th ACM SIGKDD International Conference on KDDM (2002)
3. Yang, W., Dia, J., Cheng, H., Lin, H.: Mining social networks for targeted advertising. In: 39th IEEE Hawaii International Conference on System Sciences, Hawaii (2006)
4. Zhou, D., Councill, I., Zha, H., Giles, C.: Discovering temporal communities from social network documents. In: 7th IEEE ICDM 2007, Omaha, Nebraska, USA (2007)
5. Bird, C., Gourley, A., Devanbu, P., Swaminathan, A., Gertz, M.: Mining email social networks in postgres. In: 3rd International Workshop on Mining Software Repositories (2006)
6. Lauw, H., Lim, E., Ng, T., Pang, H.: Social network discovery by mining spatio-temporal events. Computational and Mathematical Organization Theory 11, 97–118 (2005)
7. Jamali, M., Abolhassani, H.: Different aspects of social network Analysis. In: IEEE/WIC/ACM International Conference on WI 2006, Hong Kong (December 2006)

Towards Dynamic Assignment of Rights and Responsibilities to Agents (Short Version)*

Farnaz Derakhshan, Peter McBurney, and Trevor Bench-Capon

Department of Computer Science, University of Liverpool, UK
{farnazd,mcburney,tbc}@Liverpool.ac.uk

Abstract. In multi-agent systems (MASs), the concepts of organization and roles provide a useful way to structure the activities of agents and of the system, and thus provide a means to structure the design and engineering of these systems. Accordingly, most frameworks for MASs have some notion of organization such as roles and rights and responsibilities (R&Rs) of agents associated to particular roles. For many applications, the roles, rights and responsibilities of agents may change during the running of the system. In these cases, roles and/or R&Rs need to be assigned to agents dynamically at run-time. In this paper, after describing normative issues of R&Rs, dynamic issues in MAS, the issue of dynamic assignment of R&Rs of roles to agents is discussed. The paper presents the basis for our future work aimed at implementing a middleware tool for such dynamic assignment in MASs.

1 Introduction

Using the metaphor of organizational infrastructure is a very fundamental and valuable pattern for developing multiagent systems (MAS), because an organization model provides a very high–level abstraction to model. The concepts of *roles*, *rights and responsibilities of roles*, and the *interrelations of roles* are examples of such abstractions provided by an organization model. However, the emergence of open, dynamic, unpredictable and distributed environments in MASs gives rise to a need to investigate novel approaches and techniques using the organizational model.

As an example, considering the organization structure in an open MAS , as agents join to the system dynamically, their roles must be assigned to them dynamically as well. Thus, dynamic assignment of roles to agents [1] is an instance where the organizational-based approach has been proposed to reduce the complexity of MAS.

This paper presents the dynamic assignment of rights and responsibilities (R&Rs) of roles to agents as a new approach to cope with the complexity of normative MAS in open environments. In a normative MAS, agents with full autonomy to follow or violate the rules form an unpredictable and dynamic environment such that the execution of rules to impose R&Rs of the agent at each instant of time must be a runtime task. This means at runtime specific R&Rs of roles are assigned to agents at each instant of time. Such assignments are based on the represented norms of the normative MAS and dynamic triggers including the actions of the agent, actions of other agents and other environmental events.

* The full version of this paper is in: www.csc.liv.ac.uk/research/techreports/techreports.html

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 1004–1008, 2008.

2 Rights and Responsibilities of Roles

This section begins with the definition of roles and also R&Rs in multiagent systems. Then, the normative viewpoint of R&Rs of roles is explained.

The concept of ***roles*** in MAS originates from real roles in organizations. In a human organization, there are some predefined and specified roles which during the organization's lifetime different individuals might fill at different times. For instance, a supermarket as an organization has roles such as *store manager* and *sales assistant.* In MAS, roles afford the *appropriate level of abstraction* for the specification of communication between agents.

In MAS ***R&Rs*** of roles have two main features: First, each role has its own set of R&Rs, *independent from the other roles*. Roles do, however, have interrelations and contribute towards the collective objectives of the MAS. Second, the R&Rs of roles are *predefined* and *independent of the agent or individual who plays the role*.

The key elements of regulative norms. In order to introduce the normative viewpoint of R&Rs (or norms), we provide some important features of norms. According to [2], there are three types of norms: Regulative norms, Constitutive norms and Distributive norms. In the following, we mention the key elements of regulative norms which will be used in this context.

- The *addressee* [3] is the norm subject who does the act. It can be an individual, an agent, either external or internal to the system.
- The *beneficiary* is someone who benefits from the norm. For example*:"The Winner of an auction is obliged to pay the Seller the price of the item."* Winner is the addressee and Seller is the beneficiary.
- The *legal modality* [3] determines whether the norm is either an obligation (e.g. *"Everybody is obliged to pay tax."*), a prohibition (e.g. *"Seller is forbidden to place a bid."*) or a permission (e.g. "*Users are permitted to join to the auction*.").
- *The act* [3] is what the addressee is commanded, prohibited or permitted to perform. In the above example, *joining to the auction* is the act.
- *Time* [3, 5] Most norms are affected by time in different ways. The notion of time in norms can be divided into start-time, deadlines and time limits. Time notion can be attached to a norm with functions of after(t), before(t) and between(t1,t2). Some norms will be activated from a moment of time for ever. Some norms are active for a period of time and after that they will be deactivated. However, some norms are timeless and hold all the time.
- *Conditions* [5] specifies that activation/deactivation of a norm is subject to some circumstances.(e.g*"If Winner pays the price Buyer is obliged to send the item."*)

Norm Enforcement. The enforcement of norms is necessary in normative systems, because of the possibility of violations. Violations are illegal actions or states that may occur. In order to control operation in accordance with the norms, and detect and handle violations, normative systems have enforcement mechanisms to respond to the actions of agents relevant to norms should be defined. Such a plan would be a *punishment* (in the case of violation) or a *reward* (when the norms retracted).

Enforcing punishments or rewards needs some extra norms which are *distributive norms*. Such extra norms include: *Check norms* for detecting the violation and *Reaction norms:* to response to the violation is.

3 Dynamic Issues in MAS

Traditionally agent based systems dealt with well-behaved entities in reliable infrastructures and simple domains; however, currently one of the main characteristics of open MAS is their dynamic environment which is more complex. Therefore, the first step for handling such complexity would be recognizing the sources of these dynamics in open MAS. In this section, the sources of dynamism and changes are explained. Then, the next section will show the influence of these sources in the process of dynamic assignment of R&Rs to agents.

- **The Source of Dynamism.** The number of agents connected to the open system is unpredictable and may change. So, the population of agents is not fixed at design time, but may emerge at run time. Consequently the variation in the population of agents is a dynamic factor in the environment of an open MAS.
- **The Source of Changes.** Runtime changes may also influence MAS. In a normative MAS, this change may cause a rule from the represented normative system to be applied. Thus, for dynamic assignment of R&Rs it is necessary to recognize sources of changes. The major sources of changes which affect MAS are actions and environmental events. ***Agent action*** is what an agent does, which is the ability of the agent to affect its environment. Therefore if an agent performs an action, a change has occurred. ***Environmental events*** are significant occurrences or changes that the agent should respond to by some means. Here we divide events in three parts: *Action of other agents, Parameter Changes* and *Passing time.*

4 Dynamic Assignments

Management of the dynamic environment in open MAS is a complicated task. In order to cope with this complexity, the solution of dynamic assignment of roles to agents has already been proposed; the authors of [1] have described this solution and supporting methodologies.

Our aim is to provide a similar dynamic assignment that improves the management of *normative multiagent systems* that is a certain class of open MAS which include specified norms. So providing dynamic assignment of R&Rs to agents of a normative MAS is the main objective of this work.

4.1 Dynamic Assignment of Roles to Agents

As a definition, the method of dynamic assignment of roles to agents is a systematic way in which, taking account of conditions of roles, the capabilities of agents and the overall context of actions, roles are dynamically assigned to agents, by a group organizer or management system. As mentioned earlier, the idea of *dynamic assignment of roles to agents* has been previously presented and supported by some methodologies [1].

For instance, suppose when *"Mari logs into the auction system as a member"* she chooses to be a buyer in the auction session of Gold Watch. So the central management system gets her request, checks the auction session's conditions (such as *"There is an age limit of 18 for joining to this auction, because of the high price."*) and provides a history check for Mari as well. After passing the checks successfully, the role of buyer, and its accompanying rights and responsibilities, will be assigned to Mari by the management system.

4.2 Dynamic Assignment of R&Rs to Agents

Recall that dynamic assignment of rights and responsibilities to agents is proposed to improve the management of normative multiagent systems. Normative multiagent systems are multiagent systems together with normative systems in which agents can make decision whether to follow the represented norms in normative system.

The normative part of the normative MAS represents all norms that agents have to follow. Such norms indicate the obligations, permissions, prohibitions, rights and norms related to sanctions including check norms and reaction norms.

As in other multiagent systems, the concepts of ***role*** and ***rights & responsibilities*** can be used in the structure of normative MAS, so norms in normative MAS can be considered as rights and responsibilities of roles which are assigned to agents at runtime. Therefore, when at runtime a role is assigned to an agent, all the norms related to that role can be assigned to that agent. For example, in an auction system, there is a set of R&Rs for the role of *Auctioneer*. So as long as *"Ali is Auctioneer."* he should follow the whole set of norms related to the role of *Auctioneer*.

Although once the role of the agent has been allocated, all the rights and responsibilities of the agent are identified, *our approach attempts to specify and assign the specific right or responsibility of an agent at each instant of runtime*. There are two main factors in such an assignment: first, the represented norms of the normative MAS and second, the dynamic triggers including the actions of the agent or other environmental events.

From the normative viewpoint, as the rules of the normative system are conditional and time-related, a norm will be fired when the condition of the norm holds or an important time is reached. From the MAS viewpoint, the sources of dynamism and change influence the environment.

As a result, the knowledge base of the normative system also contains all conditional rules (R&Rs of roles). When a change occurs in a normative MAS, a condition of a norm may become satisfied, and the corresponding norm will be fired. We have already defined the sources of dynamism and changes as changing the population of agents, occurrence of an action or environmental events. Therefore, occurrence of any of the above sources may cause the condition of a right or a responsibility to be satisfied, so that a dynamic assignment of R&R takes place.

For example: Norm1**:** *"The Auctioneer is obliged to reject lower bids, during the auction session."* Norm2**:** *"During the auction session, if a lower bid is placed and Auctioneer did not reject it, punishment_ 2 will be applied for Auctioneer*." According to Norm1, the obligation is activated and assigned to Auctioneer agent only during the auction session. Norm2 shows that if auctioneer agent violates the auction session, s/he will be punished. So if the condition of this norm is satisfied it will be activated

and assigned to Auctioneer. As a result, the activation and deactivation of the above norm is subject to the conditions of time (during the auction), event (place a lower bid) and action (rejection of bid).

Thus the activation and deactivation of ***each specific norm*** happens dynamically at runtime. So assigning ***each activated norm*** to ***the relevant agent*** will be a dynamic task as well. In this work we aim to provide such assignment.

5 Conclusion and Future Work

The design and engineering of multi-agent systems is an important area of current research in computer science. To date, methodologies for agent-oriented software design have assumed that roles, rights and responsibilities are assigned to agents at design-time, rather than at run-time. The ability to assign roles, rights and responsibilities dynamically is important for several reasons: first, this ability increases the operational autonomy of the system. Thus, systems with this capability may have greater robustness, being able to operate effectively in a wider variety of circumstances. Second, the ability of agents to identify and punish undesirable behaviors at run-time reduces the need for system designers to identify and exclude all such behaviors at design-time. Finally, identification and punishment of undesirable behaviors may be undertaken immediately the behaviors happen.

In our future work we intend to define a normative language based on the mentioned normative issues for creating the knowledge base of R&Rs of agents. Such a knowledge base is the normative resource for the middleware tool which in the next step we will design for normative MAS. This tool will enable the practical dynamic assignment of R&Rs to agents in actual MASs.

References

1. Partsakoulakis, I., Vouros, G.: Roles in MAS: Managing the Complexity of Tasks and Environments. In: Wagner, T. (ed.) Multi-Agent Systems: An Application Science. Kluwer Academic Publisher, Dordrecht (2004)
2. Therborn, G.: Back to Norms! On the Scope and Dynamics of Norms and Normative. Action. Current Sociology 50(6), 863–880 (2002)
3. Kralingen, R.W.V., Visser, P.R.S., Bench-Capon, T.J.M., Herik, H.J.V.D.: A principled approach to developing legal knowledge systems. International Journal of Human-Computer Studies archive 51(6), 1127–1154 (1999)
4. Sartor, G.: Fundamental legal concepts: A formal and teleological characterisation. Artificial Intelligence and Law 14(1-2), 101–142 (2006)
5. Vázquez-Salceda, J., Aldewereld, H., Dignum, F.: Norms in multiagent systems:from theory to practice. J. Computer Sys. Science & Engineering CRL 20, 225–236 (2005)

Finding Correlation between Protein Protein Interaction Modules Using Semantic Web Techniques

Mehdi Kargar, Shahrouz Moaven, and Hassan Abolhassani

Department of Computer Engineering,
Sharif University of Technology, Tehran, Iran
{ Moaven,Kargar}@mehr.sharif.edu, Abolhassani@sharif.edu

Abstract. Many complex networks such as social networks and computer show modular structures, where edges between nodes are much denser within modules than between modules. It is strongly believed that cellular networks are also modular, reflecting the relative independence and coherence of different functional units in a cell. In this paper we used a human curated dataset. In this paper we consider each module in the PPI network as ontology. Using techniques in ontology alignment, we compare each pair of modules in the network. We want to see that is there a correlation between the structure of each module or they have totally different structures. Our results show that there is no correlation between proteins in a protein protein interaction network.

Keywords: Protein Protein Interaction Network, Modularity, Semantic Web.

1 Introduction

Modularity is a widespread concept in computer science, cognitive science, organization theory and other scientific and technological fields. It allows a complex system or task to be broken down into smaller, simpler functions. At the same time, the individual modules can be modified or operated on independently. Thus, changes in one part of a system should not affect other parts.

In This paper we consider the modules in a Yeast Protein Protein Interaction Network as Ontology. Ontology is one of the key concepts of Semantic Web literatures. Our purpose is to find that is there a correlation between modules in PPI. Also we want to find that is there a structural similarity between PPI modules. For this aim, we use techniques of alignment as it is used in the semantic web. Also we will consider that if two aligned protein have the same functionality or not.

2 Formal Statement of the Problem

One subject that has received an important amount of attention is the detection and characterization of community structures in networks and graphs, meaning the appearance of densely connected groups of nodes, with only sparser connections between groups. The ability to identify such groups could be of significant practical importance.

H. Sarbazi-Azad et al. (Eds.): CSICC 2008, CCIS 6, pp. 1009–1012, 2008.

For instance, groups within the internet might correspond to sets of web pages on related subjects; groups within social networks might correspond to social groups or communities. Merely the finding that a network contains tightly groups at all can convey useful data: if a cellular network were divided into such modules, for instance, it could provide evidence for a modular view of the network's dynamics and statistics, with different groups of vertices performing different functions with some sort of independence [1].

Several methods are available to separate a network into modules. Empirical and simulation results suggested that the method of Guimera and Amaral has the best outcome because it can give the most exact module separation [6]. Thus, we used this method to separate modules in the yeast protein protein interaction networks. The detail of the output is presented in the next sections.

In bioinformatics, a sequence alignment is a way of arranging the original sequences of DNA(strings of nucleotides), RNA, or protein (strings of amino acids) to identify regions of similarity that may be a consequence of functional, structural, or evolutionary relationships between the sequences. Aligned strings of nucleotide or amino acid residues are typically represented as rows within a matrix. Gaps are inserted among the residues so that residues with similar or identical characters are aligned in successive columns [2].

Ontology is a system that describes concepts and the relationships among them. Thus, what we would like to do is to produce ontology for bioinformatics and biology domain. It is important to notify that this will just be one of much possible ontology for bioinformatics. A good field of research in the area of knowledge representation has known that ontology must necessarily reflect a specific view of the knowledge. Consider, for example, the concept of protein. From a bioinformatics perspective, it is clear that the concept of an accession number should be associated with a protein. It is the key to retrieving information about a protein from sequence databases. But, it probably makes no sense to talk about accession number as a property of actual proteins in ontology built to describe the biochemistry of the cell [3].

Ontologies have been developed in the artificial intelligence community to explain a variety of fields, and have been suggested as a mechanism to provide applications with domain knowledge and to facilitate the sharing of data. The importance of ontologies has been identified within the bioinformatics community, and work has begun on developing as sharing bimolecular ontologies.

In order to support these activities successfully, the representation used for ontology should be rich enough in terms of the services it suggests, and should have a firm interpretation.

Traditionally, ontologies have been represented using static models. These can aid in exchange of data at a purely terminological or syntactic level, but can suffer because of the difficulties of interpretation. The Relationships in the model rely solely on the perspective of the modeler. If we are to share data, a clearer semantics is required. Full interaction with ontology requires, in addition, a concept of the range of services, functionally or reasoning the ontology can provide [4].

In this paper, we define ontology on the top of protein protein interaction networks. We used the interactions as the relationship between nodes. The nodes are the actual

proteins and the label of them is the sequence of amino acids. We will use this ontology to find correlation between modules in a protein protein interaction networks.

3 Method

We downloaded the PPI data for the budding yeast Saccharomyces cerevisiae from the Munich Information Center for Protein Sequences (MIPS) [5]. The dataset was human curated and included mostly binary interactions directly observed in Y2H experiments. Also, about 10% of the binary interactions in the dataset was inferred using either the spoke or matrix model from protein complexes identified by high-confidence small-scale experiments. Because it is only meaningful to separate modules within a connected part of a network, we studied the largest connected subset (i.e., the giant component), of a network. The giant component contains more than 90% of all nodes in the yeast Protein Protein Interaction network. For simplicity, we refer to the giant component of a network as the network, unless otherwise mentioned. We found 3885 number of proteins, 7260 number of interactions and 26 modules. Also the average degree is 3.74, average number of proteins per modules is 149, density ratio is 2.6 and modularity is 0.6672.

The purpose is to find out that is there a correlation between modules in a protein protein interaction network. We define an ontology on the entire network. The nodes in the ontology are proteins and the edges are the interaction between them. It should be noted that the edges are not directed. Also we assume that if protein A interacts with protein B, protein B interacts with protein A too. In the semantic web and ontology literatures, for finding similarity between nodes, there are some techniques. One the most important techniques are label similarity. For finding the similarity between labels in usual ontology, we can use a dictionary or a thesaurus. In this way, we can find the similarity between two strings. E.g. if we have car and cars, using a dictionary we can find out that these are similar string. But if we have car and automobile, we should use a thesaurus to find out that car and automobile have the same meanings. Also we can define some distance between strings and consider some strings which are beyond a threshold, as similar string. One the most famous measure for distance is hamming distance which is used widely in applications.

In our application, because our labels are the sequence of amino acids we can use a dictionary or thesaurus. Obviously, we should use some techniques such as similarity distance. Fortunately, there is a well known technique which we previously introduce. This technique is called sequence alignment. Sequence alignment is one of the most used techniques in bioinformatics. Thus, we used this technique as the similarity.

There are other techniques in ontology alignment in semantic web. These techniques are related to the structure of the ontology graph. We can use techniques in our work either. If two nodes are connected to similar nodes in the network we can assume that there are similar with each other. After applying all of these techniques, we can find out that is there similar structures among modules in the network or there is not. Also we will consider that is there a functional similarity between aligned proteins in the network or there is not. For finding out this question, we will use the famous and robust database in bioinformatics.

4 Results and Discussion

In table 1 there are the results of our method for finding correlation between modules in protein protein interaction network. We compare each duplicated and similar proteins with their pairs and compare the proteins that interact with them. In this table there are the numbers of neighbors of similar proteins which are similar too. On the other hand, we look at two similar proteins (according to our defined alignment program) and then try to find similar proteins which interact with them. By interaction we mean that there is an edge between them.

Table 1. Correlation between modules in a protein protein interaction network

Number of similar neighbors	Number of pairs
0	304
1	13
2	7
4	2

Looking at table 1, it turns out that there is not any correlation between modules of protein protein interaction network. The number of similar neighbors of two similar proteins is almost zero in each pairs and the number of nonzero similar proteins is very small. Thus there is a significant difference between similar and non similar proteins and we can conclude that there is not any correlation between modules in a protein protein interaction network.

References

1. Grivan, T., Newman, M.: Community structure in social and biological networks. PNAS, Applied Mathematics 99, 609–625 (2002)
2. Brudno, M., Malde, S.: Glocal alignment: finding rearrangements during alignment. Bioinformatics 19, 54–62 (2003)
3. Deng, S.: Mapping gene ontology to proteins based on protein-protein interaction data. Bioinformatics 20, 895–902 (2004)
4. Paton, B.: Conceptual modelling of genomic information. BMC Bioinformatics 16, 678–699 (2000)
5. Guldener, U.: MPact: The MIPS protein interaction resource on yeast. Nucleic Acids Res. 34, 436–441 (2006)
6. Guimera, R., Amaral, N.: Functional cartography of complex metabolic networks. Nature 443, 895–899 (2005)

Author Index

Abbasian, Houman 807
Abdollahi Azgomi, Mohammad 535
Abolhassani, Hassan 17, 380, 443, 867, 960, 976, 981, 1009
Afroozi Milani, Ghazal 913
Aghajani, Mohammadreza 543
Ahmadi-Noubari, Hossein 897
Ahmed, Bestoun S. 728
Akbari, Ahmad 803, 807
Akbarzadeh T., M.R. 741
Akhgari, Ehsan 871
Akhondi-Asl, Alireza 340
Akhoondi, Masoud 867
Akhtarkavan, Ehsan 130
Akramifar, Seyed Ali 857
Ali, Borhanuddin M. 153
Aliakbary, Sadegh 976
Alimardani, Fateme 884
Alipour, Hooshmand 816
Alishahi, Mohammad Mehdi 799
Alizadeh, Bijan 697
Alizadehrad, Davood 799
Alpkocak, Adil 69
Amini, Morteza 559, 862
Ansari, Ebrahim 884
Araabi, Babak Nadjar 9, 33
Arabi Nare, Somaye 518
Arabnejad, Mohammad 926
Arami, Arash 33
Asadi, Mohsen 41
Asghar Khodaparast, Ali 749, 754
Askari, Mehdi 723
Attarchi, Sepehr 364
Azadi Parand, Fereshteh 388
Azimifar, Zohreh 186
Azim Sharifloo, Amir 964
Azimzadeh, F. 396
Azizi Mazreah, Arash 679

Babaali, Bagher 485
Baghban Karimi, Ouldooz 160
Bagheri, Alireza 930
Bagheri, Mojtaba 176, 203
Bagheri Shouraki, Saeed 477, 876
Bagherzadeh, Nader 98
Bahn, Jun Ho 98
Bahrami, Saeed 852
Bahrani, Mohammad 485
Bakhshi, Marzieh 299
Bakhshi, Saeedeh 123, 299
Banaiyan, Abbas 790
Banizaman, Hamed 763
Barenco Abbas, Cláudia J. 138
Basri, ShahNor 728
Basseda, Reza 834
Bench-Capon, Trevor 1004
Berangi, Reza 718
Berenjkoub, Mehdi 585, 994
Boostani, Reza 884, 888

Cárdenas, Nelson 138
Chitforoush, Fatemeh 283
Chitsaz, Elham 1

Dadlani, Aresh 543
Dalvi, Mohammad 745
Dana, Arash 435
Darmani, Yousef 551
Dastghaibyfard, GholamHossein 799
Davari, Pooya 324, 332
Davila, Néstor 138
Davoodi, Alireza 811
Davoodi-Bojd, Esmaeil 621
Delavari, Hadi 842
Derakhshan, Farnaz 1004
Derakhshan, Nima 773
Doraisamy, Shyamala 569
Dressler, Enrico 526

Ebadi, Mojtaba 905
Esfahani, Naeem 777
Esmailee, Maryam 477
Esnaashari, M. 758

Faez, Karim 364
Fakhraie, Sied Mehdi 267, 790
Fanaei, Mohammad 994
Fanian, Ali 585, 994
Faraji-Dana, Reza 340

Farhid, M. 880
Fatemi, Hassan 17
Fatemi, Omid 194
Fathy, Mahmood 160
Fathy, Mahmoud 601
Forsati, Rana 737
Fos-hati, Amin 905
Foster, Mary Ellen 308
Fotouhi Ghazvini, M.H. 147
Froghani, Amirhossein 469
Fujita, Masahiro 697

Ganchev, Todor 493
Garakani, Vahid 917
Gardoni, Mickaël 509
Gedeon, Tom 852
Ghaemi Bafghi, Abbas 577
Ghalambor, Mohammad 786
Ghanbari, Mohammad 946
Gharehbaghi, Amir Masoud 243
Ghasemi, Hamid Reza 219
Ghasemi, Taha 275
Ghasemzadeh, Mohammad 714
Ghassem-Sani, Gholamreza 857
Ghassemi, Fatemeh 419
Ghavami, Behnam 671
Ghaznavi-Ghoushchi, Mohammad Bagher 847
Ghodsi, Mohammad 41, 283, 871
Golzari, Shahram 569
Gomrokchi, Maziar 905
Gorgin, Saeid 235

Habibi, Jafar 61, 372, 453, 960, 985
Habibnezhad Korayem, Moharram 892
Hadad, Amir H. 852
Hadj-Hamou, Khaled 509
Haghjoo, Mostafa S. 786, 917
Haghpanah, Nima 867
Haghshenas, Alireza 601
Hajhosseini, Marzieh 435
Hajimirsadeghi, G. Hossein 9
Hamidian, Hajar 340
Hashemi, Mahmoud Reza 194, 219
Hashemi Namin, Shoaleh 106
Hashempour, Masoud 705
Hassanli, Kourosh 921
Hassanpour, Hamid 324, 332
Hassan Savoji, Mohammad 773
Hassas Yeganeh, Soheil 453
Hatefi-Ardakani, Hassan 243
Hazar, Nima 901
Hendessi, Faramarz 955
Hessabi, Shaahin 90, 106, 115, 243
Homayounpour, M.M. 25
Honari, Sina 905
Hormati, Abbas 267, 790
Hosseini, Mehdi 380
Hosseinzadeh, Khosro 485
Hosseinzadeh Moghaddam, Mohammad 930

Ibrahim, H. 396
Iranmanesh, Ehsan 795, 811
Iranmanesh, Zeinab 443
Isazadeh, Ayaz 816
Isfahani, Shirin N. 732
Izadi, Mohammad 972
Izadi, Sayyed Kamyar 917

Jaberipur, Ghassem 235
Jafari, Fahimeh 990
Jafari, Shahram 926
Jafarian, Jafar Haadi 862
Jahanian, Ali 689
Jahromi, Mansoor Z. 1
Jalali, Mehrdad 838
Jalili, Rasool 427, 551, 559, 862
Javaheri, Reza 653
Javidan, Reza 186
Jooyandeh, Mohammadreza 82

Kafieh, Rahele 609
Kahani, Mohsen 577
Kamal, Mehdi 90
Kamandi, Ali 985
Kargar, Mehdi 1009
Kargar, Mohammad Javad 396
Karimpour Darav, Nima 115
Kasaei, Shohreh 176, 203, 637, 732
Kasiri-Bidhendi, Soudeh 782
Katebi, Seraj D. 1
Khadivi, Pejman 412, 585
Khadivi Heris, Hossein 897
Khalili, Amir Hossein 176, 637
Khansari, Mohammad 211
Khanteymoori, A.R. 25
Khatun, S. 153
Khayami, Raouf 909, 913
Khayatzadeh Mahani, Ali 921

Khayyamian, Mahdy 976
Khodamoradi, Kamyar 985
Khonsari, Ahmad 543, 990
Khorsandi, Siavash 593
Khosravi, Hamid 811
Khosravi, Hassan 811
Kianrad, Ahmad 543
Knoll, Alois 308
Koohi, Somayyeh 90
Krohn, Martin 404
Kurup, Gopakumar 153

Layeb, Abdesslem 942
Lee, Fong-Cheng 168
Lobalsamo, Giácomo 138
Lombardi, Fabrizio 705
Lotfi, Tayebeh 203
Lucas, Caro 834
Lucke, Ulrike 526

Mahdavi, Mehrdad 737
Mahini, Alireza 718
Mahini, Hamidreza 718
Mahmoudi Aznaveh, Ahmad 645
Malekian, Ehsan 227
Mamat, Ali 838
Manochehri, Kooroush 938
Mansouri, Azadeh 645
Manzuri Shalmani, Mohammad Taghi 130, 661, 679
Mashayekhi, Hoda 427, 960
Mashreghian Arani, Zahra 705
Masnadi-Shirazi, Mohammad A. 186
McBurney, Peter 1004
Mehdizadeh, Abbas 153
Mehdizadeh, Arash 934
Mehri, Alireza 609
Meinel, Christoph 714
Menhaj, M.B. 25
Meybodi, M.R. 758
Minagar, Sara 842
Mirian-Hosseinabadi, Seyed-Hassan 777
Mirjalily, Ghasem 968
Mirsafian, Atefeh S. 732
Moaven, Shahrouz 1009
Mobasheri, Hamid 732
Mohades, Ali 795
Mohades Khorasani, Ali 82
Mohammadi, Shahin 901
Mohammadi-nodooshan, Alireza 551
Mohi, Maryam 1000
Mohtashami, Hossein 718
Mokhtari, Maryam 601
Mokhtaripour, Alireza 745
Momeni, Mohsen 830
Momtazpour, Marjan 412
Montazeri, Allahyar 259
Montazeri, Mohammad Ali 955
Moradi, Amir 661
Moradianzadeh, Pooya 1000
Mousavi, Hamid 461
Mousavi, Mir Hashem 364
Movaghar, Ali 419, 427, 461, 737
Mowlaee, Pejman 469
Muhammad Sheikh, Noor 501
Müller, Thomas 308
Mustapha, Norwati 838

Nabaee, Mahdy 9
Naderi, Majid 921
Naghdinezhad, Amir 194
Najafi, Zahra 629
Najibi, Mehrdad 671, 689, 951
Nakhkash, Mansour 723
Nasersharif, Babak 807
Nasri, Mitra 535
Navab, Nasir 629
Navidpour, Sara 972
Nazif, Ali Nasri 795
Nazm-Bojnordi, Mahdi 267
Neshati, Mahmood 17
Nikkhah-Bahrami, Mansour 897
Niknafs, Ali 41
Nikooghadam, Morteza 227
Nikseresht, Amir Hossein 799
Noorollahi Ravari, Ali 559
Nourani, Mehrdad 551
Nourbakhsh, Azamossadat 892
Nouri, Mostafa 61
Nouri Bygi, Mojtaba 283

Omidi, Milad 773
Osareh, Alireza 356
Ozcan, Giyasettin 69

Parsa, Saeed 388, 518
Parvinnia, Elham 909, 913
Pashazadeh, Saeid 769
Pedram, Hossein 671, 951

Piri, Razieh 443
Poshtan, Javad 259
Pourmorteza, Amir 348
Pourmozaffari, Saadat 593, 938

Rabiee, Hamid R. 211, 946
Raeesi N., Mohammad R. 372
Rafati, Kamyar 777
Rafe, Vahid 291
Rahgozar, Maseud 834
Rahmani, Adel T. 291
Rahmani, Mohsen 803
Rahman Ramli, Abdul 396, 728
Rahmanzadeh, Vahid 847
Raja Abdullah, R.S.A. 147, 153
Rajabi, Ali 543
Raji, Hamid 609
Ranjbar Noiey, Abolzafl 842
Raoufi, Pariya 372
Raoufifard, Somaye 671
Rashdi, Adnan 501
Rashidi Moakhar, Ali 926
Rasid, M.F.A. 147
Rasouli, Razieh 946
Rasoulifard, Amin 577
Razzazi, Mohammad Reza 275
Rezvani, Mostafa 689
Rodman, Robert D. 493
Rokni Dezfouli, Seyyed Ali 453
Roodaki, Alireza 348, 629
Rostami, Habib 372, 960
Rostami, Mohammad Javad 749, 754
Rouhizadeh, Masoud 316

Saadat, Reza 723
Sadeghi, Mohsen 509
Sadeghian, Babak 938
Sadegh Mohammadi, Hamid Reza 493
Sadighi Moshkenani, Mohsen 1000
Sadri, Saeed 609
Saeidi, Rahim 493
Safaei, Ali A. 786
Saffarian, Amir S. 964
Saheb Zamani, Morteza 689, 934
Sahraei, Alireza 981
Saidouni, Djamel-Eddine 942
Sajadieh, Sayyed Mahdi 585
Salehi, Mostafa E. 790
Salmasizadeh, Mahmoud 661
Sameti, Hossein 485, 825
Samsudin, Khairulmizam 728
Sarbazi-Azad, Hamid 123, 299
Sardashti, Somayeh 219
Saremi, Fatemeh 427, 960
Sayadiyan, Abolghasem 469
Sayyadi, Hassan 981
Sedaghat, Reza 653
Sedaghati-Mokhtari, Naser 267
Semsarzadeh, Mehdi 219
Setarehdan, Seyed Kamaledin 629
Seyed Aghazadeh, Babak 897
Seyed Razi, Seyed Hasan 543
Shabazi, Saeed 852
Shadgar, Bita 356
Shahbazi, Hamed 745
Shams, Fereidoon 964
Shamsfard, Mehrnoush 316
Shariat, Shahriar 211
Shariati, Saeed 535
Sharifi, Mohsen 769
Sheikh, Asrar ul Haq 501
Sheikhalishahi, Seyyed Mehdi 799
Sheikh Zefreh, Mohammad 585, 955
Shih, Jung-Bin 168
Shirani, Rostam 955
Shiri, Mohammad E. 811
Shiri, Nematollaah 50
Shiry Ghidary, Saeed 782
Shoaie Shirehjini, Zahra 477
Soleymani Baghshah, Mahdieh 876
Soltanian-Zadeh, Hamid 340, 348, 621, 763
Soltanzadi, Armin 629
Soryani, M. 251
Sulaiman, Md Nasir B. 569, 838

Taheri, Mohammad 1
Tahmasebi, Mohammad 888
Taki, Arash 629
Talebi, Mohammad S. 990
Tavangarian, Djamshid 404, 526
Tayarani N., M. 741
Tinati, M.A. 880
Tofighi, Seyed Hamid Reza 348
Torkamani Azar, Farah 645
Tork Ladani, Behrouz 745

Udzir, Nur Izura 569
Unger, Helena 404

Vahabi, M. 147
Vahidi-Asl, Mojtaba 518
Veisi, Hadi 485, 825

Wang, Wen-Fong 168

Yaghmaee, Mohammad H. 990
Yang, Yixian 821
Yarmohammadi, Mahsa A. 316
Yarmohammadi, Mahshid A. 316
Yasami, Yasser 593
Yazdandoost, Maryam 283
Yazdani, Ashkan 348
Yazdian Dehkordi, Mahdi 888
Yousefi, Saleh 160
Yousefian, Nima 803
Yu, Yihua 821

Zakerolhosseini, Ali 227
Zamanifar, Kamran 830
ZamanZadeh, Sharareh 951
Zender, Raphael 526
Zheng, Shihui 821
Zheng, Zhi Hong 50
Ziaie, Alireza 871
Ziaie, Pujan 308
Ziarati, Koorush 909, 913
Zolfaghari Jooya, A. 251