

Natarajan Meghanathan
Brajesh Kumar Kaushik
Dhinaharan Nagamalai (Eds.)

Communications in Computer and Information Science

131

Advances in Computer Science and Information Technology

First International Conference on Computer Science
and Information Technology, CCSIT 2011
Bangalore, India, January 2011
Proceedings, Part I

Part 1

Communications
in Computer and Information Science

131

Natarajan Meghanathan Brajesh Kumar Kaushik
Dhinaharan Nagamalai (Eds.)

Advances in Computer Science and Information Technology

First International Conference on Computer Science
and Information Technology, CCSIT 2011
Bangalore, India, January 2-4, 2011
Proceedings, Part I

Volume Editors

Natarajan Meghanathan
Jackson State University
Jackson, MS, USA
E-mail: nmeghanathan@jsums.edu

Brajesh Kumar Kaushik
Indian Institute of Technology
Roorkee, India
E-mail: bkk23fec@iitr.ernet.in

Dhinaharan Nagamalai
Wireilla Net Solutions PTY Ltd
Melbourne, Victoria, Australia
E-mail: dhinthia@yahoo.com

Library of Congress Control Number: 2010941308

CR Subject Classification (1998): H.4, C.2, I.2, H.3, D.2, I.4

ISSN 1865-0929
ISBN-10 3-642-17856-1 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-17856-6 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

springer.com

© Springer-Verlag Berlin Heidelberg 2011
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper 06/3180

Preface

The First International Conference on Computer Science and Information Technology (CCSIT-2011) was held in Bangalore, India, during January 2–4, 2011. CCSIT attracted many local and international delegates, presenting a balanced mixture of intellect from the East and from the West.

The goal of this conference series is to bring together researchers and practitioners from academia and industry to focus on understanding computer science and information technology and to establish new collaborations in these areas. Authors are invited to contribute to the conference by submitting articles that illustrate research results, projects, survey work and industrial experiences describing significant advances in all areas of computer science and information technology.

The CCSIT-2011 Committees rigorously invited submissions for many months from researchers, scientists, engineers, students and practitioners related to the relevant themes and tracks of the workshop. This effort guaranteed submissions from an unparalleled number of internationally recognized top-level researchers. All the submissions underwent a strenuous peer-review process which comprised expert reviewers. These reviewers were selected from a talented pool of Technical Committee members and external reviewers on the basis of their expertise. The papers were then reviewed based on their contributions, technical content, originality and clarity. The entire process, which includes the submission, review and acceptance processes, was done electronically. All these efforts undertaken by the Organizing and Technical Committees led to an exciting, rich and a high-quality technical conference program, which featured high-impact presentations for all attendees to enjoy, appreciate and expand their expertise in the latest developments in computer network and communications research.

In closing, CCSIT-2011 brought together researchers, scientists, engineers, students and practitioners to exchange and share their experiences, new ideas and research results in all aspects of the main workshop themes and tracks, and to discuss the practical challenges encountered and the solutions adopted. We would like to thank the General and Program Chairs, organization staff, the members of the Technical Program Committees and external reviewers for their excellent and tireless work. We also want to thank Springer for the strong support, and the authors who contributed to the success of the conference. We sincerely wish that all attendees benefited scientifically from the conference and wish them every success in their research.

It is the humble wish of the conference organizers that the professional dialogue among the researchers, scientists, engineers, students and educators continues beyond the event and that the friendships and collaborations forged will linger and prosper for many years to come.

Natarajan Meghanathan
B.K. Kaushik
Dhinaharan Nagamalai

VIII Organization

B. Srinivasan	Monash University, Australia
Balasubramanian K.	Lefke European University, Cyprus
Boo-Hyung Lee	KongJu National University, South Korea
Chih-Lin Hu	National Central University, Taiwan
Cho Han Jin	Far East University, South Korea
Cynthia Dhinakaran	Hannam University, South Korea
Dhinaharan Nagamalai	Wireilla Net Solutions Pty Ltd., Australia
Dimitris Kotzinos	Technical Educational Institution of Serres, Greece
Dong Seong Kim	Duke University, USA
Farhat Anwar	International Islamic University, Malaysia
Firkhan Ali Bin Hamid Ali	Universiti Tun Hussein Onn Malaysia, Malaysia
Ford Lumban Gaol	University of Indonesia
Girija Chetty	University of Canberra, Australia
H.V. Ramakrishnan	MGR University, India
Henrique Joao Lopes Domingos	University of Lisbon, Portugal
Ho Dac Tu	Waseda University, Japan
Hoang, Huu Hanh	Hue University, Vietnam
Hwangjun Song	Pohang University of Science and Technology, South Korea
Jacques Demerjian	Communication & Systems, Homeland Security, France
Jae Kwang Lee	Hannam University, South Korea
Jan Zizka	SoNet/DI, FBE, Mendel University in Brno, Czech Republic
Jeong-Hyun Park	Electronics Telecommunication Research Institute, South Korea
Jivesh Govil	Cisco Systems Inc. - CA, USA
Johann Groschdl	University of Bristol, UK
John Karamitsos	University of the Aegean, Samos, Greece
Johnson Kuruvila	Dalhousie University, Halifax, Canada
Jose Enrique Armendariz-Inigo	Universidad Publica de Navarra, Spain
Jungwook Song	Konkuk University, South Korea
K.P.Thooyamani	Bharath University, India
Khoa N. Le	Griffith University , Australia
Krzysztof Walkowiak	Wroclaw University of Technology, Poland
Lu Yan	University of Hertfordshire, UK
Luis Veiga	Technical University of Lisbon, Portugal
Marco Rocchetti	University of Bologna, Italy
Michal Wozniak	Wroclaw University of Technology, Poland
Mohsen Sharifi	Iran University of Science and Technology, Iran
Murugan D.	Manonmaniam Sundaranar University, India
N. Krishnan	Manonmaniam Sundaranar University, India

Nabendu Chaki	University of Calcutta, India
Natarajan Meghanathan	Jackson State University, USA
Nidaa Abdual Muhsin Abbas	University of Babylon, Iraq
Paul D. Manuel	Kuwait University, Kuwait
Phan Cong Vinh	London South Bank University, UK
Ponpit Wongthongtham	Curtin University of Technology, Australia
Rajendra Akerkar	Technomathematics Research Foundation, India
Rajesh Kumar P.	The Best International, Australia
Rajkumar Kannan	Bishop Heber College, India
Rakhesh Singh Kshetrimayum	Indian Institute of Technology-Guwahati, India
Ramayah Thurasamy	Universiti Sains Malaysia, Malaysia
Sagarmay Deb	Central Queensland University, Australia
Sanguthevar Rajasekaran	University of Connecticut, USA
Sarmistha Neogyv	Jadavpur University, India
Sattar B. Sadkhan	University of Babylon, Iraq
Sergio Ilarri	University of Zaragoza, Spain
Serguei A. Mokhov	Concordia University, Canada
SunYoung Han	Konkuk University, South Korea
Susana Sargento	University of Aveiro, Portugal
Salah S. Al-Majeed	University of Essex, UK
Vishal Sharma	Metanoia Inc., USA
Wei Jie	University of Manchester, UK
Yannick Le Moullec	Aalborg University, Denmark
Yeong Deok Kim	Woosong University, South Korea
Yuh-Shyan Chen	National Taipei University, Taiwan
Sriman Narayana Iyengar	VIT University, India
A.P. Sathish Kumar	PSG Institute of Advanced Studies, India
Abdul Aziz	University of Central Punjab, Pakistan.
Nicolas Sklavos	Technological Educational Institute of Patras, Greece
Shivan Haran	Arizona State University, USA
Danda B. Rawat	Old Dominion University, USA
Khamish Malhotra	University of Glamorgan, UK
Eric Renault	Institut Telecom – Telecom SudParis, France
Kamaljit I. Lakhtaria	Atmiya Institute of Technology and Science, India
Andreas Riener	Johannes Kepler University Linz, Austria
Syed Rizvi	University of Bridgeport, USA
Velmurugan Ayyadurai	Center for Communication Systems, UK
Syed Rahman	University of Hawaii-Hilo, USA

Sajid Hussain	Fisk University, USA
Suresh Sankaranarayanan	University of West Indies, Jamaica
Michael Peterson	University of Hawaii at Hilo, USA
Brajesh Kumar Kaushik	Indian Institute of Technology, India
Yan Luo	University of Massachusetts Lowell, USA
Yao-Nan Lien	National Chengchi University, Taiwan
Rituparna Chaki	West Bengal University of Technology, India
Somitra Sanadhya	IIT-Delhi, India
Debasis Giri	Haldia Institute of Technology, India
S.Hariharan	B.S. Abdur Rahman University, India

Organized By



ACADEMY & INDUSTRY RESEARCH COLLABORATION CENTER (AIRCC)

Table of Contents – Part I

Distributed and Parallel Systems and Algorithms

Improved Ant Colony Optimization Technique for Mobile Adhoc Networks	1
<i>Mano Yadav, K.V. Arya, and Vinay Rishiwal</i>	
A Performance Comparison Study of Two Position-Based Routing Protocols and Their Improved Versions for Mobile Ad Hoc Networks . . .	14
<i>Natarajan Meghanathan</i>	
Privacy Preserving Naïve Bayes Classification Using Trusted Third Party Computation over Distributed Progressive Databases	24
<i>Keshavamurthy B.N. and Durga Toshniwal</i>	
Cluster Based Mobility Considered Routing Protocol for Mobile Ad Hoc Network	33
<i>Soumyabrata Saha and Rituparna Chaki</i>	
Speech Transaction for Blinds Using Speech-Text-Speech Conversions . . .	43
<i>Johnny Kanisha and G. Balakrishanan</i>	
Floating-Point Adder in Techology Driven High-Level Synthesis	49
<i>M. Joseph, Narasimha B. Bhat, and K. Chandra Sekaran</i>	
Differential Artificial Bee Colony for Dynamic Environment	59
<i>Syed Raziuddin, Syed Abdul Sattar, Rajya Lakshmi, and Moin Parvez</i>	
FAM2BP: Transformation Framework of UML Behavioral Elements into BPMN Design Element	70
<i>Jayeeta Chanda, Ananya Kanjilal, Sabnam Sengupta, and Swapan Bhattacharya</i>	
A Cross-Layer Framework for Adaptive Video Streaming over IEEE 802.11 Wireless Networks	80
<i>Santhosha Rao, M. Vijaykumar, and Kumara Shama</i>	
FIFO Optimization for Energy-Performance Trade-off in Mesh-of-Tree Based Network-on-Chip	90
<i>Santanu Kundu, T.V. Ramaswamy, and Santanu Chattopadhyay</i>	
Outliers Detection as Network Intrusion Detection System Using Multi Layered Framework	101
<i>Nagaraju Devarakonda, Srinivasulu Pamidi, Valli Kumari V., and Govardhan A.</i>	

A New Routing Protocol for Mobile Ad Hoc Networks	112
<i>S. Rajeswari and Y. Venkataramani</i>	
Remote-Memory Based Network Swap for Performance Improvement . . .	125
<i>Nirbhay Chandorkar, Rajesh Kalmady, Phool Chand, Anup K. Bhattacharjee, and B.S. Jagadeesh</i>	
Collaborative Alert in a Reputation System to Alleviate Colluding Packet Droppers in Mobile Ad Hoc Networks	135
<i>K. Gopalakrishnan and V. Rhymend Uthariaraj</i>	
Secure Service Discovery Protocols for Ad Hoc Networks	147
<i>Haitham Elwahsh, Mohamed Hashem, and Mohamed Amin</i>	
Certificate Path Verification in Peer-to-Peer Public Key Infrastructures by Constructing DFS Spanning Tree	158
<i>Balachandra, Ajay Rao, and K.V. Prema</i>	
A Novel Deadlock-Free Shortest-Path Dimension Order Routing Algorithm for Mesh-of-Tree Based Network-on-Chip Architecture	168
<i>Kanchan Manna, Santanu Chattopadhyay, and Indranil Sen Gupta</i>	
2-D DOA Estimation of Coherent Wideband Signals Using L-Shaped Sensor Array	179
<i>P.M. Swetha and P. Palanisamy</i>	
An Approach towards Lightweight, Reference Based, Tree Structured Time Synchronization in WSN	189
<i>Surendra Rahamatkar and Ajay Agarwal</i>	
Towards a Hierarchical Based Context Representation and Selection by Pruning Technique in a Pervasive Environment	199
<i>B. Vanathi and V. Rhymend Uthariaraj</i>	
An Analytical Model for Sparse and Dense Vehicular Ad hoc Networks	209
<i>Sara Najafzadeh, Norafida Ithnin, and Ramin Karimi</i>	
MIMO Ad Hoc Networks-Mutual Information and Channel Capacity . . .	217
<i>Chowdhuri Swati, Mondal Arun Kumar, and P.K. Baneerjee</i>	
Optimization of Multimedia Packet Scheduling in Ad Hoc Networks Using Multi-Objective Genetic Algorithm	225
<i>R. Muthu Selvi and R. Rajaram</i>	
TRIDENT: Isolating Dropper Nodes with Some Degree of Selfishness in MANET	236
<i>Ahmed M. Abd El-Haleem, Ihab A. Ali, Ibrahim I. Ibrahim, and Abdel Rahman H. El-Sawy</i>	

DSP/Image Processing/Pattern Recognition/ Multimedia

Physiologically Based Speech Synthesis Using Digital Waveguide Model	248
<i>A.R. Patil and V.T. Chavan</i>	
Improving the Performance of Color Image Watermarking Using Contourlet Transform	256
<i>Dinesh Kumar and Vijay Kumar</i>	
Face Recognition Using Multi-exemplar Discriminative Power Analysis	265
<i>Ganesh Bhat and K.K. Achary</i>	
Efficient Substitution-Diffusion Based Image Cipher Using Modified Chaotic Map	278
<i>I. Shatheesh Sam, P. Devaraj, and R.S. Bhuvaneshwaran</i>	
Non Local Means Image Denoising for Color Images Using PCA	288
<i>P.A. Shyji and M. Wilscy</i>	
An Iterative Method for Multimodal Biometric Face Recognition Using Speech Signal	298
<i>M. Nageshkumar and M.N. ShanmukhaSwamy</i>	
Quality Analysis of a Chaotic and Hopping Stream Based Cipher Image	307
<i>G.A. Sathishkumar, K. Bhoopathybagan, and N. Sriraam</i>	
Design Pattern Mining Using State Space Representation of Graph Matching	318
<i>Manjari Gupta, Rajwant Singh Rao, Akshara Pande, and A.K. Tripathi</i>	
MST-Based Cluster Initialization for K-Means	329
<i>Damodar Reddy, Devender Mishra, and Prasanta K. Jana</i>	
Geometry and Skin Color Based Hybrid Approach for Face Tracking in Colour Environment	339
<i>Mahesh Goyani, Gitam Shikkenawis, and Brijesh Joshi</i>	
Performance Analysis of Block and Non Block Based Approach of Invisible Image Watermarking Using SVD in DCT Domain	348
<i>Mahesh Goyani and Guvantsinh Gohil</i>	
A Novel Topographic Feature Extraction Method for Indian Character Images	358
<i>Soumen Bag and Gaurav Harit</i>	

Comprehensive Framework to Human Recognition Using Palmprint and Speech Signal	368
<i>Mahesh P.K. and ShanmukhaSwamy M.N.</i>	

Logical Modeling and Verification of a Strength Based Multi-agent Argumentation Scheme Using NuSMV	378
<i>Shravan Shetty, H.S. Shashi Kiran, Murali Babu Namala, and Sanjay Singh</i>	

Key Frame Detection Based Semantic Event Detection and Classification Using Heirarchical Approach for Cricket Sport Video Indexing	388
<i>Mahesh M. Goyani, Shreyash K. Dutta, and Payal Raj</i>	

Software Engineering

Towards a Software Component Quality Model	398
<i>Nitin Upadhyay, Bharat M. Despande, and Vishnu P. Agrawal</i>	

Colour Image Encryption Scheme Based on Permutation and Substitution Techniques	413
<i>Narendra K. Pareek, Vinod Patidar, and Krishan K. Sud</i>	

Deciphering the Main Research Themes of Software Validation – A Bibliographical Study	428
<i>Tsung Teng Chen, Yaw Han Chiu, and Yen Ping Chi</i>	

Toward a Comprehension View of Software Product Line	439
<i>Sami Ouali, Naoufel Kraiem, and Henda Ben Ghezala</i>	

Collecting the Inter Component Interactions in Multithreaded Environment	452
<i>Arun Mishra, Alok Chaurasia, Pratik Bhadkoliya, and Arun Misra</i>	

On the Benefit of Quantification in AOP Systems – A Quantitative and a Qualitative Assessment	462
<i>Kotrappa Sirbi and Prakash Jayanth Kulkarni</i>	

Empirical Measurements on the Convergence Nature of Differential Evolution Variants	472
<i>G. Jeyakumar and C. Shanmugavelayutham</i>	

Database and Data Mining

An Intelligent System for Web Usage Data Preprocessing	481
<i>V.V.R. Maheswara Rao, V. Valli Kumari, and K.V.S.V.N. Raju</i>	

Discovery and High Availability of Services in Auto-load Balanced Clusters	491
<i>Shakti Mishra, Alok Mathur, Harit Agarwal, Rohit Vashishtha, D.S. Kushwaha, and A.K. Misra</i>	
Bayes Theorem and Information Gain Based Feature Selection for Maximizing the Performance of Classifiers.....	501
<i>Subramanian Appavu, Ramasamy Rajaram, M. Nagammai, N. Priyanga, and S. Priyanka</i>	
Data Mining Technique for Knowledge Discovery from Engineering Materials Data Sets	512
<i>Doreswamy, K.S. Hemanth, Channabasayya M. Vastrad, and S. Nagaraju</i>	
Mobile Database Cache Replacement Policies: LRU and PRRRP	523
<i>Hariram Chavan and Suneeta Sane</i>	
A Novel Data Mining Approach for Performance Improvement of EBGM Based Face Recognition Engine to Handle Large Database.....	532
<i>Soma Mitra, Suparna Parua, Apurba Das, and Debasis Mazumdar</i>	
Efficient Density Based Outlier Handling Technique in Data Mining	542
<i>Krishna Gopal Sharma, Anant Ram, and Yashpal Singh</i>	
Hierarchical Clustering of Projected Data Streams Using Cluster Validity Index	551
<i>Bharat Pardeshi and Durga Toshniwal</i>	
Non-Replicated Dynamic Fragment Allocation in Distributed Database Systems	560
<i>Nilarun Mukherjee</i>	
Soft Computing (AI, Neural Networks, Fuzzy Systems, etc.)	
A Comparative Study of Machine Learning Algorithms as Expert Systems in Medical Diagnosis (Asthma).....	570
<i>B.D.C.N. Prasad, P.E.S.N. Krishna Prasad, and Yeruva Sagar</i>	
Stabilization of Large Scale Linear Discrete-Time Systems by Reduced Order Controllers	577
<i>Sundarapandian Vaidyanathan and Kavitha Madhavan</i>	
Hybrid Synchronization of Hyperchaotic Qi and Lü Systems by Nonlinear Control.....	585
<i>Sundarapandian Vaidyanathan and Suresh Rasappan</i>	

An Intelligent Automatic Story Generation System by Revising Proppian's System	594
<i>Jaya A. and Uma G.V.</i>	
A Genetic Algorithm with Entropy Based Probabilistic Initialization and Memory for Automated Rule Mining	604
<i>Saroj, Kapila, Dinesh Kumar, and Kanika</i>	
Author Index	615

Table of Contents – Part II

Networks and Communications

Analysis of Successive Interference Cancellation in CDMA Systems	1
<i>G.S. Deepthy and R.J. Susan</i>	
Scenario Based Performance Analysis of AODV and DSDV in Mobile Adhoc Network	10
<i>S. Taruna and G.N. Purohit</i>	
Web Service Based Sheltered Medi Helper	20
<i>Priya Loganathan, Jeyalakshmi Jeyabalan, and Usha Sarangapani</i>	
Reliable Neighbor Based Multipath Multicast Routing in MANETs	33
<i>Rajashekhhar C. Biradar and Sunilkumar S. Manvi</i>	
Ad-Hoc On Demand Distance Vector Routing Algorithm Using Neighbor Matrix Method in Static Ad-Hoc Networks	44
<i>Aitha Nagaraju, G. Charan Kumar, and S. Ramachandram</i>	
Improving QoS for Ad-Hoc Wireless Networks Using Predictive Preemptive Local Route Repair Strategy	55
<i>G.S. Sharvani, T.M. Rangaswamy, Aayush Goel, B. Ajith, Binod Kumar, and Manish Kumar</i>	
Improving Energy Efficiency in Wireless Sensor Network Using Mobile Sink	63
<i>K. Deepak Samuel, S. Murali Krishnan, K. Yashwant Reddy, and K. Suganthi</i>	
Transformed Logistic Block Cipher Scheme for Image Encryption	70
<i>I. Shatheesh Sam, P. Devaraj, and R.S. Bhuvaneswaran</i>	
Context Management Using Ontology and Object Relational Database (ORDBMS) in a Pervasive Environment	79
<i>B. Vanathi and V. Rhymend Uthariaraj</i>	
Path Optimization and Trusted Routing in MANET: An Interplay between Ordered Semirings	88
<i>Kiran K. Somasundaram and John S. Baras</i>	
LEAD: Energy Efficient Protocol for Wireless Ad Hoc Networks	99
<i>Subhankar Mishra, Sudhansu Mohan Satpathy, and Abhipsa Mishra</i>	

Scale-Down Digital Video Broadcast Return Channel via Satellite (DVB-RCS) Hub	107
<i>N.G. Vasantha Kumar, Mohanchur Sarkar, Vishal Agarwal, B.P. Chaniara, S.V. Mehta, V.S. Palsule, and K.S. Dasgupta</i>	
Reconfigurable Area and Speed Efficient Interpolator Using DALUT Algorithm	117
<i>Rajesh Mehra and Ravinder Kaur</i>	
Performance Evaluation of Fine Tuned Fuzzy Token Bucket Scheme for High Speed Networks	126
<i>Anurag Aeron, C. Rama Krishna, and Mohan Lal</i>	
A Survey on Video Transmission Using Wireless Technology	137
<i>S.M. Koli, R.G. Purandare, S.P. Kshirsagar, and V.V. Gohokar</i>	
Cluster Localization of Sensor Nodes Using Learning Movement Patterns	148
<i>R. Arthi and K. Murugan</i>	
Improving TCP Performance through Enhanced Path Recovery Notification	158
<i>S. Sathya Priya and K. Murugan</i>	
The Replication Attack in Wireless Sensor Networks: Analysis and Defenses	169
<i>V. Manjula and C. Chellappan</i>	
New Distributed Initialization Protocols for IEEE 802.11 Based Single Hop Ad Hoc Networks	179
<i>Rasmeet S. Bali and C. Rama Krishna</i>	
Deployment of GSM and RFID Technologies for Public Vehicle Position Tracking System	191
<i>Apurv Vasal, Deepak Mishra, and Puneet Tandon</i>	
Type of Service, Power and Bandwidth Aware Routing Protocol for MANET	202
<i>Divyanshu, Ruchita Goyal, and Manoj Mishra</i>	
Energy Efficiency in Wireless Network Using Modified Distributed Efficient Clustering Approach	215
<i>Kaushik Chakraborty, Abhrajit Sengupta, and Himadri Nath Saha</i>	
ERBR: Enhanced and Improved Delay for Requirement Based Routing in Delay Tolerant Networks	223
<i>Mohammad Arif, Kavita Satija, and Sachin Chaudhary</i>	

Deterministic Approach for the Elimination of MED Oscillations and Inconsistent Routing in BGP	233
<i>Berlin Hency, Vasudha Venkatesh, and Raghavendran Nedunchezhian</i>	
Context Aware Mobile Initiated Handoff for Performance Improvement in IEEE 802.11 Networks	243
<i>Abhijit Sarma, Shantanu Joshi, and Sukumar Nandi</i>	
Safety Information Gathering and Dissemination in Vehicular Ad hoc Networks: Cognitive Agent Based Approach	254
<i>M.S. Kakkasageri and S.S. Manvi</i>	
Enhanced AODV Routing Protocol for Mobile Adhoc Networks	265
<i>K.R. Shobha and K. Rajanikanth</i>	
A Comparative Study of Feedforward Neural Network and Simplified Fuzzy ARTMAP in the Context of Face Recognition	277
<i>Antu Annam Thomas and M. Wilscy</i>	
A Multicast-Based Data Dissemination to Maintain Cache Consistency in Mobile Environment	290
<i>Kahkashan Tabassum and A. Damodaram</i>	
Quantitative Analysis of Dependability and Performability in Voice and Data Networks.....	302
<i>Almir P. Guimarães, Paulo R.M. Maciel, and Rivalino Matias Jr.</i>	
Rain Fade Calculation and Power Compensation for Ka-Band Spot Beam Satellite Communication in India	313
<i>Jayadev Jena and Prasanna Kumar Sahu</i>	
Generalized N X N Network Concept for Location and Mobility Management	321
<i>C. Ashok Baburaj and K. Alagarsamy</i>	
A Weight Based Double Star Embedded Clustering of Homogeneous Mobile Ad Hoc Networks Using Graph Theory.....	329
<i>T.N. Janakiraman and A. Senthil Thilak</i>	
An Enhanced Secured Communication of MANET	340
<i>J. Thangakumar and M. Robert Masillamani</i>	
MIMO-OFDM Based Cognitive Radio for Modulation Recognition	349
<i>R. Deepa and K. Baskaran</i>	
Network and Communications Security	
Mutual Authentication of RFID Tag with Multiple Readers	357
<i>Selwyn Piramuthu</i>	

Design Space Exploration of Power Efficient Cache Design Techniques	362
<i>Ashish Kapania and H.V. Ravish Aradhya</i>	
Secured WiMAX Communication at 60 GHz Millimeter-Wave for Road-Safety	372
<i>Bera Rabindranath, Sarkar Subir Kumar, Sharma Bikash, Sur Samarendra Nath, Bhaskar Debasish, and Bera Soumyasree</i>	
A Light-Weight Protocol for Data Integrity and Authentication in Wireless Sensor Networks	383
<i>Jibi Abraham, Nagasimha M P, Mohnish Bhatt, and Chaitanya Naik</i>	
Intrusion Detection Using Flow-Based Analysis of Network Traffic.....	391
<i>Jisa David and Ciza Thomas</i>	
A Threshold Cryptography Based Authentication Scheme for Mobile Ad-hoc Network	400
<i>Haimabati Dey and Raja Datta</i>	
Formal Verification of a Secure Mobile Banking Protocol	410
<i>Huy Hoang Ngo, Osama Dandash, Phu Dung Le, Bala Srinivasan, and Campbell Wilson</i>	
A Novel Highly Secured Session Based Data Encryption Technique Using Robust Fingerprint Based Authentication	422
<i>Tanmay Bhattacharya, Sirshendu Hore, Ayan Mukherjee, and S.R. Bhadra Chaudhuri</i>	
An Active Host-Based Detection Mechanism for ARP-Related Attacks	432
<i>F.A. Barbhuiya, S. Roopa, R. Ratti, N. Hubballi, S. Biswas, A. Sur, S. Nandi, and V. Ramachandran</i>	
Privacy-Preserving Naïve Bayes Classification Using Trusted Third Party Computation over Vertically Partitioned Distributed Progressive Sequential Data Streams	444
<i>Keshavamurthy B.N. and Durga Toshniwal</i>	
Copyright Protection in Digital Multimedia	453
<i>Santosh Kumar, Sumit Kumar, and Sukumar Nandi</i>	
Trust as a Standard for E-Commerce Infrastructure	464
<i>Shabana and Mohammad Arif</i>	
A Scalable Rekeying Scheme for Secure Multicast in IEEE 802.16 Network	472
<i>Sandip Chakraborty, Soumyadip Majumder, Ferdous A. Barbhuiya, and Sukumar Nandi</i>	

Secure Data Aggregation in Wireless Sensor Networks Using Privacy Homomorphism	482
<i>M.K. Sandhya and K. Murugan</i>	
Deniable Encryption in Replacement of Untappable Channel to Prevent Coercion	491
<i>Jaydeep Howlader, Vivek Nair, and Saikat Basu</i>	
Author Identification of Email Forensic in Service Oriented Architecture	502
<i>Pushendra Kumar Pateriya, Shivani Mishra, and Shefalika Ghosh Samaddar</i>	
A Chaos Based Approach for Improving Non Linearity in S Box Design of Symmetric Key Cryptosystems	516
<i>Jeyamala Chandrasekaran, B. Subramanyan, and Raman Selvanayagam</i>	
Implementation of Invisible Digital Watermarking by Embedding Data in Arithmetically Compressed Form into Image Using Variable-Length Key	523
<i>Samanta Sabyasachi and Dutta Saurabh</i>	
Wireless and Mobile Networks	
Transmission Power Control in Virtual MIMO Wireless Sensor Network Using Game Theoretic Approach	535
<i>R. Valli and P. Dananjayan</i>	
Protocols for Network and Data Link Layer in WSNs: A Review and Open Issues	546
<i>Ankit Jain, Deepak Sharma, Mohit Goel, and A.K. Verma</i>	
Psychoacoustic Models for Heart Sounds	556
<i>Kiran Kumari Patil, B.S. Nagabhushan, and B.P. Vijay Kumar</i>	
Location Based GSM Marketing	564
<i>Sparsh Arora and Divya Bhatia</i>	
Inverse Square Law Based Solution for Data Aggregation Routing Using Survival Analysis in Wireless Sensor Networks	573
<i>Khaja Muhaiyadeen A., Hari Narayanan R., Shelton Paul Infant C., and Rajesh G.</i>	
System Implementation of Pushing the Battery Life and Performance Information of an Internal Pacemaker to the Handheld Devices via Bluetooth Version 3.0 + H.S	584
<i>Balasundaram Subbusundaram and Gayathri S.</i>	

Cross-Layer Analyses of QoS Parameters in Wireless Sensor Networks	595
<i>Alireza Masoum, Nirvana Meratnia, Arta Dilo, Zahra Taghikhaki, and Paul J.M. Havinga</i>	
Optimum Routing Approach vs. Least Overhead Routing Approach for Minimum Hop Routing in Mobile Ad hoc Networks	606
<i>Natarajan Meghanathan</i>	
Architecture for High Density RFID Inventory System in Internet of Things	617
<i>Jain Atishay and Tanwer Ashish</i>	
A Relative Study of MANET and VANET: Its Applications, Broadcasting Approaches and Challenging Issues	627
<i>Ajit Singh, Mukesh Kumar, Rahul Rishi, and D.K. Madan</i>	
Adaptive Threshold Based on Group Decisions for Distributed Spectrum Sensing in Cognitive Adhoc Networks	633
<i>Rajagopal Sreenivasan, G.V.K. Sasirekha, and Jyotsna Bapat</i>	
A Distributed Mobility-Adaptive and Energy Driven Clustering Algorithm for MANETs Using Strong Degree	645
<i>T.N. Janakiraman and A. Senthil Thilak</i>	
Dynamic Sink Placement in Wireless Sensor Networks	656
<i>Parisa D. Hossein Zadeh, Christian Schlegel, and Mike H. MacGregor</i>	
Author Index	667

Table of Contents – Part III

Soft Computing (AI, Neural Networks, Fuzzy Systems, etc.)

Analysis of the Severity of Hypertensive Retinopathy Using Fuzzy Logic	1
<i>Aravinthan Parthibarajan, Gopalakrishnan Narayanamurthy, Arun Srinivas Parthibarajan, and Vigneswaran Narayanamurthy</i>	
An Intelligent Network for Offline Signature Verification Using Chain Code	10
<i>Minal Tomar and Pratibha Singh</i>	
An Improved and Adaptive Face Recognition Method Using Simplified Fuzzy ARTMAP	23
<i>Antu Annam Thomas and M. Wilscy</i>	
An Automatic Evolution of Rules to Identify Students' Multiple Intelligence	35
<i>Kunjil Mankad, Priti Srinivas Sajja, and Rajendra Akerkar</i>	
A Survey on Hand Gesture Recognition in Context of Soft Computing	46
<i>Ankit Chaudhary, J.L. Raheja, Karen Das, and Sonia Raheja</i>	
Handwritten Numeral Recognition Using Modified BP ANN Structure	56
<i>Amit Choudhary, Rahul Rishi, and Savita Ahlawat</i>	
Expert System for Sentence Recognition	66
<i>Bipul Pandey, Anupam Shukla, and Ritu Tiwari</i>	
RODD: An Effective Reference-Based Outlier Detection Technique for Large Datasets	76
<i>Monowar H. Bhuyan, D.K. Bhattacharyya, and J.K. Kalita</i>	

Distributed and Parallel Systems and Algorithms

A Review of Dynamic Web Service Composition Techniques	85
<i>Demian Antony D'Mello, V.S. Ananthanarayana, and Supriya Salian</i>	
Output Regulation of Arneodo-Couillet Chaotic System	98
<i>Sundarapandian Vaidyanathan</i>	

A High-Speed Low-Power Low-Latency Pipelined ROM-Less DDFS	108
<i>Indranil Hatai and Indrajit Chakrabarti</i>	
A Mathematical Modeling of Exceptions in Healthcare Workflow	120
<i>Sumagna Patnaik</i>	
Functional Based Testing in Web Services Integrated Software Applications	130
<i>Selvakumar Ramachandran, Lavanya Santapoor, and Haritha Rayudu</i>	
Design and Implementation of a Novel Distributed Memory File System	139
<i>Urvashi Karnani, Rajesh Kalmady, Phool Chand, Anup Bhattacharjee, and B.S. Jagadeesh</i>	
Decentralized Dynamic Load Balancing for Multi Cluster Grid Environment	149
<i>Malarvizhi Nandagopal and V. Rhymend Uthariaraj</i>	
Adoption of Cloud Computing in e-Governance	161
<i>Rama Krushna Das, Sachidananda Patnaik, and Ajita Kumar Misro</i>	
Efficient Web Logs Stair-Case Technique to Improve Hit Ratios of Caching	173
<i>Khushboo Hemnani, Dushyant Chawda, and Bhupendra Verma</i>	
A Semantic Approach to Design an Intelligent Self Organized Search Engine for Extracting Information Relating to Educational Resources . . .	183
<i>B. Saleena, S.K. Srivatsa, and M. Chenthil Kumar</i>	
Cluster Bit Collision Identification for Recognizing Passive Tags in RFID System	190
<i>Katheerja Parveen, Sheik Abdul Khader, and Munir Ahamed Rabbani</i>	
Integration Testing of Multiple Embedded Processing Components	200
<i>Hara Gopal Mani Pakala, K.V.S.V.N. Raju, and Ibrahim Khan</i>	
 Security and Information Assurance	
A New Defense Scheme against DDoS Attack in Mobile Ad Hoc Networks	210
<i>S.A. Arunmozhi and Y. Venkataramani</i>	
A Model for Delegation Based on Authentication and Authorization	217
<i>Coimbatore Chandrasekaran and William R. Simpson</i>	
Identification of Encryption Algorithm Using Decision Tree	237
<i>R. Manjula and R. Anitha</i>	

A Novel Mechanism for Detection of Distributed Denial of Service Attacks	247
<i>Jaydip Sen</i>	
Authenticating and Securing Mobile Applications Using Microlog	258
<i>Siddharth Gupta and Sunil Kumar Singh</i>	
Assisting Programmers Resolving Vulnerabilities in Java Web Applications	268
<i>Pranjal Bathia, Bharath Reddy Beerelli, and Marc-André Laverdière</i>	
Estimating Strength of a DDoS Attack Using Multiple Regression Analysis	280
<i>B.B. Gupta, P.K. Agrawal, R.C. Joshi, and Manoj Misra</i>	
A Novel Image Encryption Algorithm Using Two Chaotic Maps for Medical Application	290
<i>G.A. Sathishkumar, K. Bhoopathyagan, N. Sriraam, SP. Venkatachalam, and R. Vignesh</i>	
Chest X-Ray Analysis for Computer-Aided Diagnostic	300
<i>Kim Le</i>	
Overcoming Social Issues in Requirements Engineering	310
<i>Selvakumar Ramachandran, Sandhyarani Dodda, and Lavanya Santapoor</i>	
Ad Hoc and Ubiquitous Computing	
Range-Free Localization for Air-Dropped WSNs by Filtering Neighborhood Estimation Improvements	325
<i>Eva M. García, Aurelio Bermúdez, and Rafael Casado</i>	
Evolution of Various Controlled Replication Routing Schemes for Opportunistic Networks	338
<i>Hemal Shah and Yogeshwar P. Kosta</i>	
Collaborative Context Management and Selection in Context Aware Computing	348
<i>B. Vanathi and V. Rhymend Uthariaraj</i>	
Privacy Preservation of Stream Data Patterns Using Offset and Trusted Third Party Computation in Retail-Shop Market Basket Analysis	358
<i>Keshavamurthy B.N. and Durga Toshniwal</i>	

Wireless Ad Hoc Networks and Sensor Networks

Application of Euclidean Distance Power Graphs in Localization of Sensor Networks	367
<i>G.N. Purohit, Seema Verma, and Usha Sharma</i>	
A New Protocol to Secure AODV in Mobile AdHoc Networks	378
<i>Avinash Krishnan, Aishwarya Manjunath, and Geetha J. Reddy</i>	
Spelling Corrector for Indian Languages	390
<i>K.V.N. Sunitha and A. Sharada</i>	
Voltage Collapse Based Critical Bus Ranking	400
<i>Shobha Shankar and T. Ananthapadmanabha</i>	
Multiplexer Based Circuit Synthesis with Area-Power Trade-Off	410
<i>Sambhu Nath Pradhan and Santanu Chattopadhyay</i>	
Exergaming – New Age Gaming for Health, Rehabilitation and Education	421
<i>Ankit Kamal</i>	
Inclusion/Exclusion Protocol for RFID Tags	431
<i>Selwyn Piramuthu</i>	
Min Max Threshold Range (MMTR) Approach in Palmprint Recognition	438
<i>Jyoti Malik, G. Sainarayanan, and Ratna Dahiya</i>	
Power and Buffer Overflow Optimization in Wireless Sensor Nodes	450
<i>Gauri Joshi, Sudhanshu Dwivedi, Anshul Goel, Jaideep Mulherkar, and Prabhat Ranjan</i>	
Web Log Data Analysis and Mining	459
<i>L.K. Joshila Grace, V. Maheswari, and Dhinaharan Nagamalai</i>	
Steganography Using Version Control System	470
<i>Vaishali S. Tidake and Sopan A. Talekar</i>	
Author Index	481

Improved Ant Colony Optimization Technique for Mobile Adhoc Networks

Mano Yadav¹, K.V. Arya², and Vinay Rishiwal³

¹ ITS Engineering College, Greater Noida, UP, India

² ABV-IIITM, Gwalior, India

³ MJP Rohilkhand University, Bareilly, UP, India

mano425@gmail.com, kvarya@iiitm.ac.in, vrishiwal@mjpru.ac.in

Abstract. Efficient routing is a crucial issue in Mobile Adhoc Networks. This paper proposes an improved algorithm for routing in mobile adhoc networks based on the Ant Colony Optimization (ACO) technique. The proposed improved ACO (I-ACO) performs routing by making use of transition probability among nodes along with available pheromone update information in ACO principle. This approach reduces the cost of ant agents. I-ACO has two phases *viz* route discovery and route maintenance and also utilizes the concept of back-tracking when the packets reaches destination node.

Simulation results show that I-ACO achieves better packet delivery ratio and reduces the average end-to-end-delay as compare to its counterpart AODV. This scheme can be incorporated in any version of the on demand routing protocol. In this paper it has been implanted in the basic AODV protocol.

Keywords: Ant Colony Algorithm, AODV, Mobile Adhoc Networks.

1 Introduction

Mobile Ad-hoc network (MANET) [1] [2] is an infrastructure less network consisting of mobile nodes. These nodes constantly change the topologies and communicate via a wireless medium. The transmission range of the nodes in MANETs is usually limited [3], so nodes within a network need to reply data packets over several intermediate nodes to communicate. Thus nodes in MANETs act as both hosts and routers. Since the nodes in MANETs are mobile, designing an appropriate routing technique for MANETs needs to be flexible enough to adapt to arbitrarily changing network topologies, and to support efficient bandwidth and energy management, since low powered batteries operate the nodes [4].

ACO [6] [7] make use of computational intelligence [5] and simulates the behaviour of ant colonies in nature as they forage for food and find the most efficient routes from their nests to food sources. The ACO technique for routing in MANET uses stigmergic [11] process to determine the best possible routes from a source node to a destination node. Artificial ants are placed at each node and they mark their trails with pheromone as they move within the network. The level of concentration of pheromone on a trail is a reflection of its quality [8] [9]. For multiple objectives [10] more than one colony can be maintained which interact to share the information gained and hence, give those paths which lie in optimal range.

ACO research has not been used much in MANETs and a good review of ACO application is given in [6]. In Early applications ACO was applied for routing in Telecommunication networks and a successful algorithm in this domain is Ant Net [11]. Existing literature [1] [2] [12] [13] [14] presents the idea of traditional and ant colony based routing in MANETs.

In this work we present an improved optimization technique for mobile adhoc networks using ACO concept. Rest of the paper is organized as follows: Section 2 presents the related work. In section 3 application of Ant Colony Optimization algorithm on MANETs is discussed. The Model Formulation is discussed in section 4. Simulation results are presented in section 5 and in section 6 conclusion and direction for future work are given.

2 Related Work

The concept of ant colony optimization is explained by Dorigo and Stutzle [6] and a hypercube framework for ACO is presented by Blum et al. [7]. Zuan et al. [15] have presented an improved adaptive ant colony routing algorithm (ARAAI) for MANETs. Their routing algorithms are based on ant colony optimization to find routing paths that are close to the shortest paths even if the nodes in the network have different transmission ranges. The drawback of these algorithms is the long delay due to large number of messages needs to be sent for path establishment. Kamali and Opatrny [16] have proposed POSANT, a reactive routing algorithm for mobile ad hoc networks which combines the idea of ant colony optimization with nodes position information. They claimed that POSANT has a shorter route establishment time while using a smaller number of control messages than other ant colony routing algorithms. Wang et al. [17] have proposed a hybrid routing algorithm (HOPNET) for MANETs based on the framework of ACO and zone routing [18]. HOPNET is based on ants hopping from one zone to the next, consists of the local proactive route discovery within a node's neighborhood and reactive communication between the neighborhoods. The algorithm has features extracted from Zone Routing [18] and DSR [1] protocols and this algorithm has been compared to AODV [20] and AntHocNet [21] routing protocols. HOPNET is highly scalable for large networks compared to AntHocNet. Cauvery and Vishwanatha [8] have given proposal is to improve the performance of the Ant colony algorithm [22]. This algorithm is on-demand source initiated routing algorithm. This is based on the principles of swarm intelligence. The algorithm is adaptive, scalable and favors load balancing. The improvement suggested in [8] handles loss of ants and resource reservation.

3 Applying ACO on MANETs

The principles of ACO routing can be used to develop a suitable solution routing in MANETs. Here an agent is used which is referred as Ant. Initially Ant is forwarded to explore all possible paths of the network using flooding and then the same Ant backtracks to establish the path information acquired leaving a pheromone amount on each node. Data packets are stochastically transmitted towards nodes with higher pheromone concentration along the path to the destination.

The network under consideration is represented as a connected graph $G = (V, E)$, with $|V| = N$ nodes. The metric of optimization can be a parameter defined by the user which may depend upon factors like number of hops, bandwidth, delay and other network conditions. The network consists of mainly two data structures (i) routing table and (ii) visited array. Each node in the network has a routing table whose size is the degree of the node times number of nodes in the network. Each pair (row, column) in the routing table has two values *viz* the pheromone concentration and the desirability of the neighbor.

Each ant maintains an array which contains the node id, of visited nodes in searching the path to the destination. The goal is to avoid the loops in the path traced by the ant while forwarding and if a loop is found during path searching then the corresponding ant is discarded.

In this paper the routing algorithm carries out three main activities namely route discovery, route maintenance and packet forwarding. The route discovery phase in the network consists of two phases forwarding and backtracking. A source node sends ants to all its neighbors in search of the destination node and the number of ants is equal to the out-degree of the source node. Each ant maintains its own visited array of nodes (say visit array) to remember its path. On an intermediate node an Ant divides itself into the number of Ants which is equal to out-degree. If an intermediate node is in the visit array of the ant then that path is discarded and further propagation of ant is stopped, otherwise that node is added to the visit array of that ant.

After reaching the destination, the ant backtracks the path according to its visit array. Pheromone value is updated in the routing table of the node according to the quality of that path. In backtracking, if a link breakage is found ant retraces the path to destination to decrease the pheromone concentration at the nodes in visit array with the same amount and that path is discarded at destination. The complete process is detailed in algorithm 1 and algorithm 2.

Algorithm 1: FORWARDING

1. At source node do the following
 - 1.1 create an ant
 - 1.2 add node to visit array
 - 1.3 Broadcast to all neighbors.
2. At intermediate node do the following
 - 2.1 IF node exists in visited array Then
discard the path and stop that ant.
Else add the node in visit array.
 - 2.2 Maintain the information about quality of path.
 - 2.3 Forward ant to all neighbours.

Algorithm 2: BACKTRACKING

1. At destination node
 - 1.1 Backtrack the path according to the entries in the visit array.
 - 1.2 IF a link breakage found Then

- 1.2.1 Decrease pheromone concentration with the same amount on each backtracked node.
 - 1.2.2 Decrement desirability counter of that node
 - Else
 - 1.2.3 IF desirability > threshold Then
 - set desirability equal to initial value
 - Else increment desirability counter by one.
- Update pheromone concentration at each node backtrack according to quality of the path.

Forwarding of packets

Established routes are maintained in the routing table. When a packet is to be delivered to a destination node, source node computes transition probability for each neighbor using desirability and pheromone concentration. The node with maximum probability is chosen as the next node for forwarding the packet. The process is described in Algorithm 3.

Algorithm 3: PACKET FORWARDING

1. Identify source node and destination node.
2. At each node
 - 2.1 Compute transition probability for all neighbors (see 3.1).
 - 2.2 Find the neighbor with maximum transition probability, set it as next node.
 - 2.3 Repeat steps 2.1 to 2.2 till packet reaches to destination.

Route Maintenance

Hello message to detect and monitor the links to neighbours is periodically broadcast to its neighbours by each node. If a node fails to receive Hello messages from a neighbor a link breakage is detected and its routing table is updated by deleting the information about that neighbor. The procedure is given in Algorithm 4.

Algorithm 4: ROUTE MAINTENANCE

1. Each node broadcasts hello message to all its neighbors.
 - If a link breakage is detected
 - Then update routing table with deleting entries for that neighbor.
 - Else update link parameter information.
2. Evaporate pheromone concentration periodically.

4 The Model Formulation

The model formulation on different parameters used in the development of the proposed algorithm is formulated as follows:

4.1 Transition Probability

Transition probability [] primarily helps in finding location of the new pixel in the movement of ants and is defined as

$$P_{i,j} = \frac{(\tau_{ij}^{\alpha})(\eta_{ij}^{\beta})}{\sum(\tau_{ij}^{\alpha})(\eta_{ij}^{\beta})}$$

Where,

τ_1, τ_2 represents the pheromone matrix element (x,y) for constraint 1 and 2 respectively, similarly η_1, η_2 , respectively presents the desirability of node (x,y) for constraint 1 and 2.

α and β are the parameters which control the influence of τ & η respectively. Here constraints are factors which restrict the movement on predefined area. In this simulation value of α is taken as 7 and β as 3.

4.2 Directional Probability

The directional probability [] increases the chances to choose the next node that is in the direction of destination. The movement in east-west direction is decided by,

East-west (ew) direction,

$$ew = (x_2 - x) * constt / penalty[k][x][y]$$

Similarly north-south direction is given by,

$$ns = (y_2 - y) * constt / penalty[k][x][y]$$

$$xlim[k] = \begin{cases} xlim[k] - ns, & x = s, se, sw; \text{ where } ns < 0 \\ xlim[k] + ns, & x = n, ne, nw; \text{ where } ns < 0 \\ xlim[k] - ew, & x = n, nw, sw; \text{ where } ew < 0 \\ xlim[k] + ew, & x = n, ne, se; \text{ where } ew \geq 0 \end{cases}$$

Where directions are represented by letters as, n: north, e: east, s: south, w: west, ne: north-east, se: south-east, nw: north-west, sw: south-west; and k is particular constraint; x : x-coordinate, y : y-coordinate, x_2 : x-coordinate of destination, y_2 : y-coordinate of destination; and $xlim[k]$ determine directional probability in direction x for constraint k.

4.3 Pheromone Update

$$\tau_{x,y} = \rho \tau_{x,y} + \Delta \tau_{x,y}$$

where,

$\tau_{x,y}$, amount of pheromone on a node (x,y) for each constraint

ρ , rate of pheromone evaporation

$\Delta \tau_{x,y}$, amount of pheromone deposited, typically given by

$$\Delta\tau_{x,y}^k = \begin{cases} \frac{1}{L_k}, & \text{if pixel } (x,y) \text{ is visited} \\ 0, & \text{otherwise} \end{cases}$$

where, L_k is the cost of the k^{th} ant's tour.

5 Simulation Framework and Results

Proposed algorithm I-ACO has been evaluated using Network Simulator (NS-2) [19]. Network test beds have been setup for 25, 50, 75 and 100 nodes in an area of $1000 * 1000 \text{ m}^2$. The performance of the proposed work (I-ACO) has been measured for packet delivery ratio and average end-to-end-delay (in seconds) by varying the values pause time from 0 to 150 (seconds) and speed of node movement from 5 to 20 (meter/second) for different network scenarios. Details of the simulation framework have been given in Table 1.

5.1 Average End to End Delay

Average end-to-end delay is the delay experienced by the successfully delivered packets in reaching their destinations. It denotes the efficiency of the underlying routing technique as delay primarily depends on optimality of path chosen.

Scenario 1: pause time 0-100 seconds, number of nodes: 50, Speed: 15 m/sec

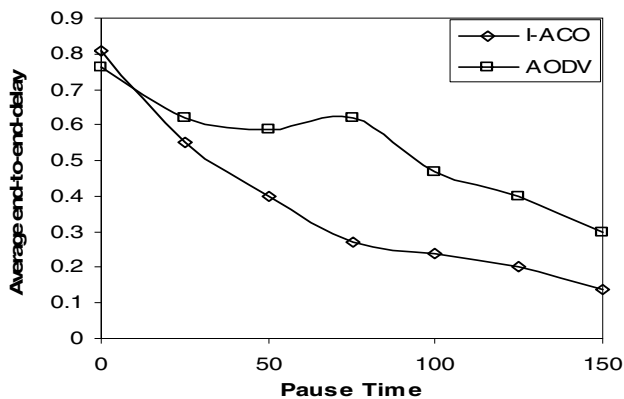
I-ACO has been compared to AODV in terms of average end-to-end-delay by varying pause time from 0 seconds to 150 seconds while keeping the speed of the nodes constant at 15 m/sec. The variation of End-to-End-Delay with respect to pause time is shown in Figure 2. It is clear from Figure 2 that initially (for 0 seconds pause time) AODV takes less time as compare to I-ACO. This is due to the initial calculations involved for setting up a route in I-ACO. As the pause time increases the average delay in both the schemes (I-ACO and AODV) is decreasing, but for all the pause times greater than 0 (zero), I-ACO exhibits less average delay as compare to its counterpart AODV.

Scenario 2: Number of Nodes: 25, 50, 75 and 100, pause time: 50 seconds, speed: 15 m/sec

The comparison of average end-to-end-delay for I-ACO and AODV with varying number of nodes (25 to 100) while keeping the value of pause time (50 seconds) and speed (15 m/sec) constant is shown in Figure 3. It is observed from Figure 3 that for any size of network from 25 nodes to 100 nodes, I-ACO bear very less delay as compare to AODV.

Table 1. Simulation Scenarios and Parameter Values for I-ACO and AODV

Parameter	Value
Simulation Time	900 seconds
Mobility Model	Random Way Point Model, Pause Time (0-150 seconds) Speed (5-20 m/sec)
MAC Protocol	802.11
Routing protocol	AODV
Network Size (Nodes)	25-100
Propagation Model	Two Ray Ground
Time Between Retransmitted requests	500 ms
Size of Source Header Carrying an Address	$4n + 4$ Bytes
Timeout for Non Propagating Search	30 ms
Maximum Rate for Sending Replies from a Route	1/s
Node Transmission Range (TRange)	150-300 m.
Terrain Area	1000 x 1000 m ²
A to control the influence of r	7
B to control the influence of η	3
P (rate of evaporation of pheromone content)	$\sqrt{\text{pheromone content}}$

**Fig. 2.** Pause Time Vs End-to-end-delay (for 50 nodes)

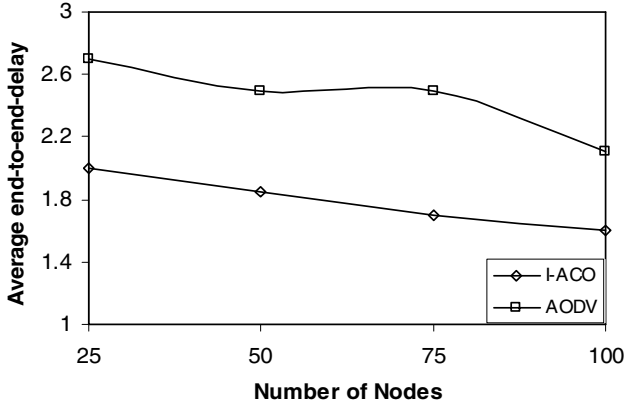


Fig. 3. Number of Nodes Vs end-to-end-delay

Scenario 3: Number of nodes: 50, speed: 15 m/sec, pause time: 50 seconds

In scenario 3, I-ACO has been compared to AODV keeping pause time and number of nodes constant while varying the speed from 5 m/sec to 20 m/sec and the effect has been shown in Figure 4. It can be observed in Figure 4 that for all the values of speed from 5 m/sec to 20 m/sec both the scheme bears more delay as the speed increases. But for all the values of speed the average end-to-end-delay for I-ACO is much less as compared to AODV.

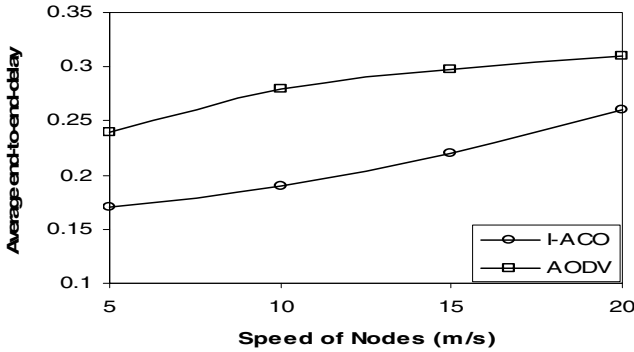


Fig. 4. Speed Vs end-to-end-delay

Scenario 4: Number of nodes: 50, speed: 15 m/sec, pause time: 50 seconds, TRange: 150m-300 m

Scenario 4 is a special case of three previous scenarios. In this scenario the value of pause time, speed and number of nodes is kept constant and the transmission range of the participating nodes has been varied from 150 meters to 300 meters and average end-to-end-delay in case of both the schemes viz I-ACO and AODV has been observed.

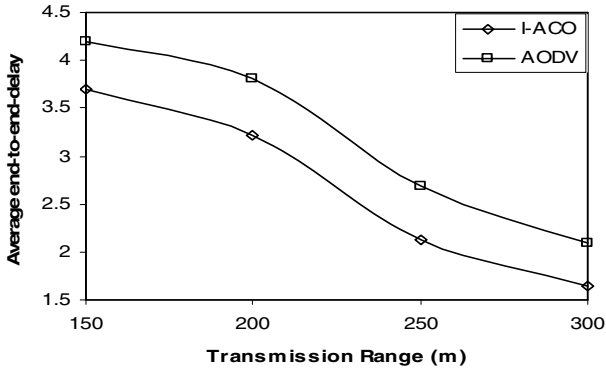


Fig. 5. Transmission range Vs end-to-end-delay

It can be seen in Figure 5 that as the transmission range of the nodes increases the value of the average end-to-end-delay decreases for both the schemes. But once again I-ACO is a better option.

5.2 Packet Delivery Ratio

PDF is defined as the fraction of successfully received packets, which survive while finding their destination. PDF determines the efficiency of the protocol to predict a link breakage and also the efficacy of the local repair process to find an alternate path. The completeness and correctness of the routing protocol is also determined.

Scenario 1: pause time 0-150 seconds, number of nodes: 50, Speed: 15 m/sec

As shown in Figure 6, I-ACO has been compared to AODV in terms of packet delivery ratio for varying pause time from 0 seconds to 150 seconds while keeping the speed of the nodes constant i.e. 15 m/sec. It can be seen in the figure 6 that, for all the values of pause time from 0 seconds to 150 seconds, packet delivery ratio increases as the value of the pause time increases. But for all the pause times greater than zero I-ACO is able to deliver more number of packets as compared to AODV. Even for larger value of pause time (150 seconds), I-ACO is able to deliver almost 99% packets successfully while AODV is delivering only 97.32 % of packets for the same value of pause time.

Scenario 2: Number of Nodes: 25, 50, 75 and 100, pause time: 50 seconds, speed: 15 m/sec

Figure 7 shows a comparison of I-ACO and AODV for packet delivery ratio for varying number of nodes (25 to 100) keeping the value of pause time (50 seconds) and speed (15 m/sec) constant. Figure 3 witnesses that for any size of network from 25 nodes to 100 nodes I-ACO deliver more number of packets successfully as compare to AODV. For the largest network of 100 nodes I-ACO is delivering more than 98% of the packets successfully while AODV is delivering only 94% of the packets for the same size of the network.

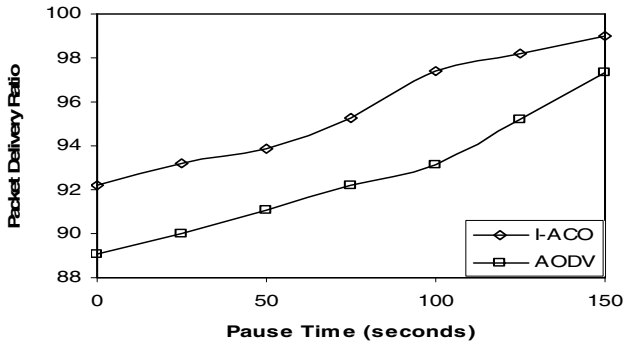


Fig. 6. Packet delivery ration Vs. pause Time

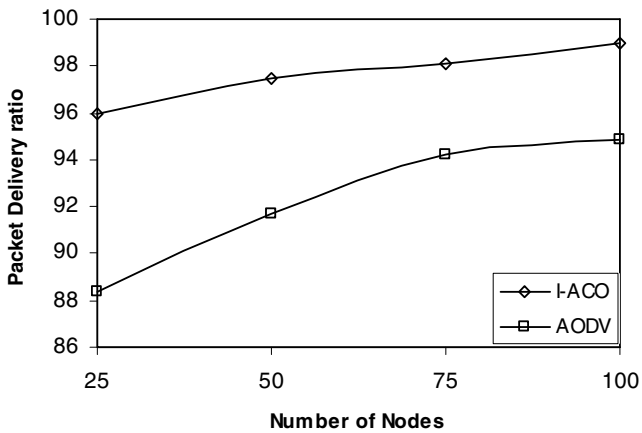


Fig. 7. Number of Nodes vs. packet delivery ratio

Scenario 3: Number of nodes: 50, speed: 15 m/sec, pause time: 50 seconds

In scenario 3, I-ACO has been compared to AODV keeping pause time and number of nodes constant while varying the value of speed from 5 m/sec to 20 m/sec and the effect has been shown in Figure 8. It can be observed in Figure 8 that for all the values of speed, I-ACO delivers more packets successfully than AODV. Even at a very high speed of 20 m/sec I-ACO delivers approximately 94% packets successfully while AODV can deliver only 82% packets.

Scenario 4: Number of nodes: 50, speed: 15 m/sec, pause time: 50 seconds, TRange: 150m-300 m

In Scenario 4, the value for pause time, speed and number of nodes is kept constant. Only the transmission range of the participating nodes has been varied from 150 meters to 300 metes and the effect of this factor has been observed on packet delivery ratio in case of both the schemes viz I-ACO and AODV.

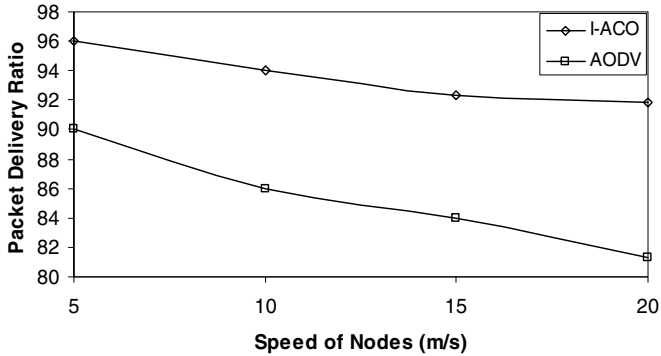


Fig. 8. Speed Vs. Packet delivery ratio

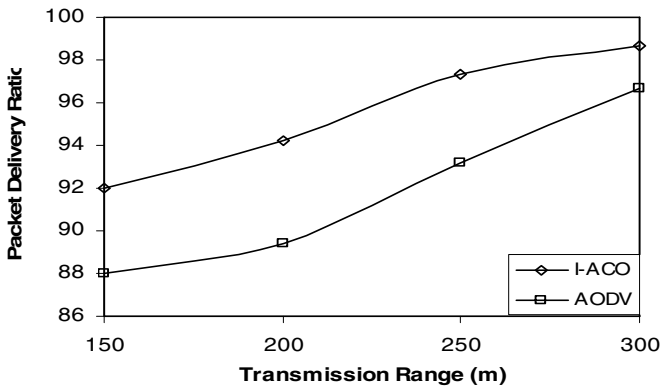


Fig. 9. Transmission range Vs. packet delivery ratio

It can be seen in Figure 9 that, increased transmission range of the nodes increases the value of the packet delivery ratio for both the schemes. But once again I-ACO is a better option in terms of number of packets successfully for different values of transmission range of the nodes from 150 meters to 300 meters. At a very high transmission range of 300 meters I-ACO is able to deliver more than 98% of the packets successfully while AODV can deliver approximately 96%.

6 Conclusion and Future Work

In this paper an improved Ant Colony Algorithm (I-ACO) has been proposed to search optimal paths in MANETs. In the proposed work an approach for routing in MANETs has been given and evaluated. The algorithm is very dynamic and robust. It incorporates mobility which is a key feature to making it adaptable to an Ad-hoc environment. The algorithm is also able to perform route maintenance and handle link failure in network. The simulation results show that the algorithm is able to achieve

better packet delivery ratio and bears less average end-to-end-delay as compare to AODV for the different values of pause time, speed, network size and transmission range. The scheme analysis is still on for checking the values of control overheads occurred for different network scenarios as well as I-ACO needs a better approach for energy management in the network.

References

1. Abolhasan, M., Wysocki, T., Dutkiewicz, E.: A Review of Routing Protocols for Mobile Adhoc Networks. *Adhoc Networks* 2(1), 1–22 (2004)
2. Chenna Reddy, P., Chandrasekhar Reddy, P.: Performance Analysis of Adhoc Network Routing Protocols. In: *International Symposium on Ad Hoc and Ubiquitous Computing (ISAUHC 2006)*, pp. 186–187 (2006)
3. Chen, Y., Siren, E.G., Wicker Stephan, B.: On selection of Optimal Transmission Power for Adhoc Network. In: *Proceedings HICCS 2003*, vol. 9, pp. 300–309 (2003)
4. Doshi, S., Brown, T.X.: Minimum Energy Routing Schemes for a Wireless Adhoc Networks. In: *Proceedings of IEEE INFOCOM*, pp. 103–111 (2002)
5. Engelbrecht, A.P.: *Computational Intelligence-An Introduction*, 2nd edn. John Wiley & Sons, Chichester (2007)
6. Dorigo, M., Stutzle, T.: *Ant Colony Optimization*. MIT Press, Cambridge (2004)
7. Blum, C., Roli, A., Dorigo, M.: The hyper-cube framework for Ant Colony Optimization. *IEEE Trans. on Systems, Man, and Cybernetics – Part B* 34(2), 1161–1172 (2004)
8. Cauvery, N.K., Vishwanatha, K.V.: Enhanced Ant Colony Based Algorithm for Routing in Mobile Ad-hoc Network. *World Academy of Science, Engineering and Technology*, 30–35 (2008)
9. Osagie, E., Thulasiraman, P., Thulasiram, R.K.: PACONET: Improved Ant Colony Optimization Routing Algorithm for mobile ad-hoc networks. In: *International Conference on Advance Information Networking and Applications (AINA 2008)*, pp. 204–211 (2008)
10. Dorigo, M., Maniozzo, V., Colomi, A.: The Ant System: Optimization by a colony of cooperating agents. *IEEE Trans. on System, Man, and Cybernetics, Part-B* 26(1), 29–41 (1996)
11. Di Caro, G., Dorigo, M.: AntNET: Distributed Stigmergetic Control Communications Network. *Journal of Artificial Research*, 317–365 (1997)
12. Gutjahr, W.J.: A Graph Based Ant System and its Convergence. *Future Generation Computer Systems* 16(9), 873–888 (2000)
13. Dorigo, M., et al.: Review of Ant colony Optimization. *Journal of Artificial Intelligence* 165(2), 261–264 (2005)
14. Fu, Y., Wang, X., Shi, W., Li, S.: Connectivity Based Greedy Algorithm with Multipoint Relaying for Mobile Ad Hoc Networks. In: *The 4th International Conference on Mobile Ad-hoc and Sensor Networks, MSN 2008*, pp. 72–76 (2008)
15. Yuan-yuan, Z., Yan-xiang, H.: Ant routing algorithm for mobile ad-hoc networks based on adaptive improvement. In: *Proceedings, International Conference on Wireless Communications, Networking and Mobile Computing*, vol. 2, pp. 678–681 (2005)
16. Kamali, S., Opatry, J.: POSANT: A Position Based Ant Colony Routing Algorithm for Mobile Ad-hoc Networks. In: *Third International Conference on Wireless and Mobile Communications, ICWMC*, p. 21. (2007)

17. Wang, J., Osagie, E., Thulasiraman, P., Thulasiram, R.K.: HOPNET: A hybrid ant colony optimization routing algorithm for mobile ad hoc network. *Journal of Adhoc Networks* 7(4), 690–705 (2009)
18. Haas, Z.J., Pearlman, M.R., Samar, P.: The Zone Routing Protocol (ZRP) for Ad Hoc Networks, IETF Internet Draft, draft-ietf-manet-zone-zrp-04.txt (July 2002)
19. Fall, K., Varadhan, K. (eds.): The ns Manual, The VINT Project. Collaboration between researchers at UC Berkeley, LBL, USC/ISI, and Xerox PARC (November 2007)
20. Perkins, C.E., Royer, E.M.: Ad Hoc On-Demand Distance Vector Routing. In: *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications*, pp. 90–100 (1999)
21. De Gianni, C., Frederick, D., Maria Gambardella, L.: AntHocNet: An ant-based hybrid routing algorithm for mobile ad hoc networks. In: *Proc. International conference on parallel problem solving from nature*, pp. 461–470 (2004)
22. Cauvery, N.K., Viswanatha, K.V.: Ant Algorithm for Mobile Ad Hoc network. In: *Proceedings of the International Conference on Advanced Computing and Communication Technologies for High Performance Applications* (2008)

A Performance Comparison Study of Two Position-Based Routing Protocols and Their Improved Versions for Mobile Ad Hoc Networks

Natarajan Meghanathan

Jackson State University
Jackson, MS 39217, USA

natarajan.meghanathan@jsums.edu

Abstract. The high-level contribution of this paper is a detailed simulation based analysis of the performance of two well-known position-based routing protocols - Greedy Perimeter Stateless Routing (GPSR) and the Geographical Routing Protocol based on Prediction (GRPP) and their improved versions to handle perimeter forwarding. The two strategies adopted to improve the performance of position-based routing protocols and better handle perimeter forwarding are: Destination-node Location Prediction (DNP) and Advanced Greedy Forwarding (AGF) approaches. We use a scalable location service scheme, referred as Hierarchical Location Service (HLS) to periodically disseminate the location information of the nodes in the network. The simulations were conducted in ns-2 under different conditions of network density, node mobility and offered traffic load. Performance results indicate that with a slightly larger location service overhead, the improved versions of GPSR and GRPP based on DNP and AGF yield a relatively lower hop count, end-to-end delay per data packet and a larger packet delivery ratio.

Keywords: Mobile Ad hoc Networks, Position-based Routing, Greedy Perimeter Stateless Routing, Geographical Routing Protocol based on Prediction, Simulations.

1 Introduction

A mobile ad hoc network (MANET) is a dynamic distributed system of wireless nodes that move independently of each other. The operating transmission range of the nodes is limited and as a result, MANET routes are often multi-hop in nature. Any node in a MANET can become a source or destination, and each node can function as a router, forwarding data for its peers. MANET routing protocols can be classified into topology-based and position-based protocols. Topology-based protocols are either proactive or reactive in nature. Proactive routing protocols determine and maintain routes between any pair of nodes irrespective of their requirement. The reactive on-demand routing protocols determine a route only when required. As the network topology changes dynamically, reactive on-demand routing has been preferred over proactive routing [3][6].

Position-based routing protocols do not conduct on-demand route discovery to learn and maintain routes. Instead, forwarding decisions are taken independently for each data packet at every forwarding node (including the source) depending on the position of the forwarding node, the intermediate nodes and the destination. Normally, the source includes its estimated location information of the destination in every data packet. The position-based routing protocols are mostly designed to choose the intermediate forwarding nodes that lie on the shortest path or close to the shortest path from the source to the destination. Greedy Perimeter Stateless Routing (GPSR) [7] and Geographical Routing Protocol based on Prediction (GRPP) [4] are two well-known examples of position-based routing protocols. Each node is assumed to know the locations of its neighbors through periodic beacon exchange. The effectiveness of the position-based routing protocols depends on the accuracy of the destination location information included in the header of the data packets, method adopted for disseminating location information and the method adopted to learn the latest location information of the destination node. In this research work, we assume the source uses the Hierarchical Location Service (HLS) [8], a robust and scalable location service scheme proposed for MANETs. The source node queries the responsible cells (in a region of the network) of the HLS at a certain time period uniformly distributed in the range $[0 \dots MaxT_{update}]$ where $MaxT_{update}$ is the maximum time period between successive destination location update searches. The source node includes the recently learnt location co-ordinates of the destination in the header of the data packets. The shorter the time between consecutive location update searches by the source node, the more accurate will be the location prediction and shorter will be the hop count. But, this advantage comes at the cost of a higher cellcast (broadcast within the responsible cells) control message overhead in frequently querying the HLS about the latest location of the destination. Due to the limited queue size at the nodes, a higher control message overhead to query HLS can also result in dropping of the data packets.

The rest of the paper is organized as follows: In Section 2, we provide a brief overview of the position-based GPSR and GRPP routing protocols and their improved versions. Section 3 presents the simulation conditions used to compare the GPSR and GRPP routing protocols and their improved versions and explains the simulation results observed for different conditions of network density, node mobility and offered traffic load. Section 4 draws the conclusions.

2 Review of Position-Based Routing Protocols

Position-based routing protocols do not go through a network-wide route discovery process, but attempt to forward the data packets from a source to the destination using the position information of the destination included in the data packet headers and the knowledge of the nodes about the positions of other nodes in their local neighborhood. Two well-known examples of position-based routing protocols are the Greedy Perimeter Stateless Routing (GPSR) protocol [7] and the Geographical Routing Protocol based on Prediction (GRPP) [4].

Greedy Perimeter Stateless Routing (GPSR) [7] is a position-based ad hoc routing protocol in which there is no flooding-based route discovery to determine source-destination routes. The source periodically uses a location service scheme (like HLS)

to learn about the latest location information of the destination and includes it in the header of every data packet. If the destination is not directly reachable, the source node forwards the data packet to the neighbor node that lies closest to the destination. Such a greedy procedure of forwarding the data packets is also repeated at the intermediate nodes. In case, a forwarding node could not find a neighbor that lies closer to the destination than itself, the node switches to perimeter forwarding. With perimeter forwarding, the data packet is forwarded to the first neighbor node that is come across, when the line connecting the forwarding node and the destination of the data packet is rotated in the anti-clockwise direction. The location of the forwarding node in which greedy forwarding failed (and perimeter forwarding began to be used) is recorded in the data packet. We switch back to greedy forwarding when the data packet reaches a forwarding node which can find a neighbor node that is away from the destination node by a distance smaller than the distance between the destination node and the node at which perimeter forwarding began. GPSR requires each node periodically (for every one second, in this paper) broadcast a beacon containing its latest location information to its neighbors.

The perimeter forwarding approach of GPSR has been observed to generate wasteful loops when the destination node moves away from the location co-ordinates included in the header of the data packets [10]. To counter the looping problem and to increase the packet delivery ratio when the destination node moves out of its original location, a destination node-location prediction (DNP) approach has been proposed in [10]. According to DNP, each node, before forwarding a data packet based on the location information of the destination in the packet header, searches its neighbor list for the destination node. If the destination node is in the neighbor list, then the data packet is directly forwarded to the destination node. A further advanced improvement to position-based greedy forwarding called Advanced Greedy Forwarding (AGF) has been proposed in [9]. According to AGF, each node manages to collect the list of nodes in its two-hop neighborhood through the exchange of neighbor lists during periodic beacon broadcast in the one-hop neighborhood. In AGF, each node, before forwarding a data packet based on the location information of the destination in the data packet header, searches for the destination in its one-hop and two-hop neighbor lists. If the destination is in the one-hop neighbor list, the data packet is directly forwarded to the destination (in this case AGF reverts to DNP). If the destination is only in the two-hop neighbor list and not in the one-hop neighbor list, the data packet is forwarded to the neighbor node (of the destination) in the one-hop neighbor list.

The Geographical Routing Protocol based on Prediction (GRPP) [4] is a novel approach of deciding the next hop node at a forwarding node, based on the current and future positions of the forwarding node and its neighboring nodes with respect to the ultimate destination node of the data packet. Based on its own movement and the periodic beacons received from its neighbors, each node learns the location of itself and its neighbors at the current time instant t (say, in seconds) and predicts the location of itself and its neighbors for the next 3 seconds (i.e., at time instants $t+1$, $t+2$ and $t+3$). Let I be the intermediate node from which a data packet needs to be forwarded so that it can reach its ultimate destination D . For every neighbor node N of I , I computes the distances d_{IN}^t , d_{ND}^t , d_{IN}^{t+1} , d_{ND}^{t+1} , d_{IN}^{t+2} , d_{ND}^{t+2} , d_{IN}^{t+3} , d_{ND}^{t+3} between itself and N and between N and D for the current time instant t and for each of the next three seconds. The location of the destination D is assumed to be fixed for the current time

instant and for each of the next three seconds. Only the location of the forwarding node I and its neighbors are predicted. The forwarding node I chooses the next hop as the neighbor node N such that the sum $d_{IN}^t + d_{ND}^t + d_{IN}^{t+1} + d_{ND}^{t+1} + d_{IN}^{t+2} + d_{ND}^{t+2} + d_{IN}^{t+3} + d_{ND}^{t+3}$ is minimized. GRPP is an improvement over the Ellipsoid algorithm [11] that considers only minimizing the sum $d_{IN}^t + d_{ND}^t$ while choosing the next hop node at the intermediate forwarding node I . Both GRPP and the Ellipsoid protocols aim to select the next hop node as the node that lies closer to the straight line joining the forwarding node and the destination. By also considering the predicted locations of the neighbor nodes to determine the next hop, GRPP selects stable links (links having positive increase in the signal strength between consecutively transmitted packets) in the presence of node mobility. Like the original version of GPSR, GRPP also relies on the co-ordinates of the destination location information in the data packet header to apply the above described procedure to determine the next hop.

In this paper, we study the original versions of GPSR and GRPP and also apply DNP and AGF to improve the performance of both GPSR and GRPP. The improved versions are referred to as GPSR_DNP, GPSR_AGF, GRPP_DNP and GRPP_AGF.

3 Simulation Environment and Performance Results

We use ns-2 (version 2.28) [5] as the simulator for our study. We implemented the GPSR, GRPP protocols and their improved versions GPSR_DNP, GPSR_AGF, GRPP_DNP and GRPP_AGF in ns-2. The network dimension used is a 1000m x 1000m square network. The transmission range of each node is assumed to be 250m. The number of nodes used is 25 and 50 nodes representing networks of low and high density respectively. Initially, nodes are uniformly randomly distributed in the network. The maximum time period ($MaxT_{update}$) between successive location update queries sent by a source node to the Hierarchical Location Service for the position-based routing protocols is chosen to be 20, 80 and 200 seconds. For a given value of $MaxT_{update}$, the time period between two successive queries launched by the source node towards the responsible cells (of the HLS) for the latest information about the destination location is uniformly distributed within the range $[0 \dots MaxT_{update}]$.

Traffic sources are constant bit rate (CBR). The number of source-destination ($s-d$) sessions used is 15 (indicating low traffic load) and 30 (indicating high traffic load). The starting timings of these $s-d$ sessions are uniformly distributed between 1 to 50 seconds. The sessions continue until the end of the simulation time, which is 1000 seconds. Data packets are 512 bytes in size and the packet sending rate is 4 data packets/second. For each node, we made sure that the node does not end up a source for more than two sessions and/ or not as a destination for more than two sessions. The node mobility model used in all of our simulations is the Random Waypoint model [1]; the values of the maximum velocity of a node, v_{max} , used are 5, 30 and 50 m/s representing scenarios of low, moderate and high node mobility respectively.

We measure the following performance metrics for the routing protocols under each of the above simulation conditions. Each data point in Figures 1 through 8 is an average of data collected using 5 mobility trace files and 5 sets of randomly selected 15 or 30 $s-d$ sessions, depending on the simulation condition.

- *Control Messages Overhead*: It is the sum of the destination location update search cellcast messages received by the nodes in the network, computed over all the s - d sessions of a simulation run.
- *Hop count per path*: It is the average hop count per path, time-averaged over all the s - d sessions.
- *End-to-end delay per packet*: It is the average of the delay incurred by the data packets that originate at the source and delivered at the destination.
- *Packet Delivery Ratio*: It is the ratio of data packets delivered to the destination to the data packets originated at the source, computed over all s - d sessions.

3.1 Control Message Overhead

We measure the control message overhead as the number of destination location update search messages received by the nodes in the network. Note that we measure the control message overhead as the number of control messages received by the nodes in the network, rather than the number of control messages transmitted. This is because the control traffic is broadcast in nature.

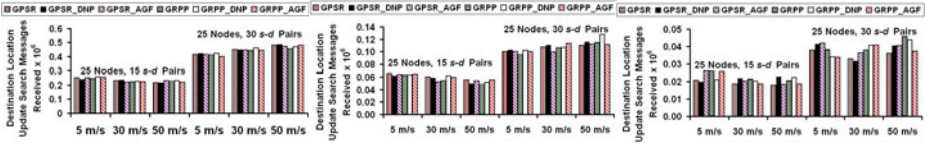


Fig. 1.1. $MaxT_{upd} = 20$ sec Fig. 1.2. $MaxT_{upd} = 80$ sec Fig. 1.3. $MaxT_{upd} = 200$ sec

Fig. 1. Control Message Overhead (25 Node Network)

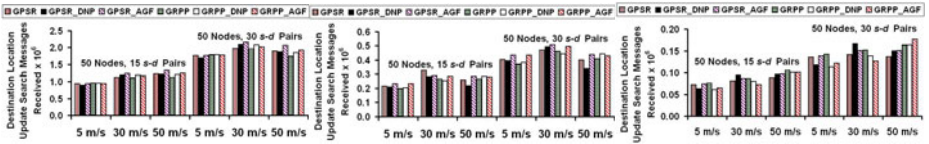


Fig. 2.1. $MaxT_{upd} = 20$ sec Fig. 2.2. $MaxT_{upd} = 80$ sec Fig. 2.3. $MaxT_{upd} = 100$ sec

Fig. 2. Control Message Overhead (50 Node Network)

The control messages considered for performance measurement are the destination location search cellcast messages in the responsible cells in the case of GPSR, GRPP, GPSR_DNP, GPSR_AGF, GRPP_DNP and GRPP_AGF. The control message overhead incurred by the position-based routing protocols (illustrated in Figures 1 and 2) depends on the value of $MaxT_{update}$ and the number of s - d pairs. For a given number of s - d pairs, the lower the value of $MaxT_{update}$, the larger the control message overhead and more accurate is the destination location information included by the source in the header of the data packets. Similarly, for a given value of $MaxT_{update}$, the larger the number of s - d pairs, more source nodes would initiate destination location searches and larger is the control message overhead. We observe a tradeoff between the control

message overhead and the percentage of packets delivered. For a given offered data traffic load, the smaller the $MaxT_{update}$ value, the larger will be the percentage of packets delivered with the position-based routing protocols and vice-versa (refer Figures 7 and 8 for packet delivery ratio).

We do not take into account the periodic beacon exchange overhead incurred while measuring the routing control message overhead. These routing protocols require periodic beacon exchange among neighbors in order to learn about the positions and mobility of the neighbor nodes. In this paper, we have set the beacon exchange interval to be 1 second. The reason for the omission is that we want to only consider control messages whose scope for transmission and reception is larger, i.e., either within the cells of a HLS region or network-wide. Though we do not measure the number of beacons received at the nodes, we do take into consideration the presence of beacons and other control messages in the queues of the nodes and thus consider their impact on packet delivery ratio and end-to-end delay per data packet.

In the case of GRPP, GPSR and their improved versions, after each destination location update search, each source node, independently, uniformly and randomly selects a waiting time value from the range $[0 \dots MaxT_{update}$ seconds] and initiates the next destination location search process. By doing so, we avoid the situation of having all nodes simultaneously and periodically broadcasting (cellcasting) their location query message updates, which would trigger congestion at the nodes.

3.2 Hop Count per Path

In Figures 3 and 4, we observe that the hop count incurred by GPSR and its improved versions is smaller than that GRPP and its improved versions. As the routing protocols simulated in this paper do not take the queue size into consideration while determining the routes, the hop count of the routes for each protocol is independent of the offered data traffic load. The hop count of the routes chosen by GPSR and GRPP are somewhat influenced by the dynamics of node mobility.

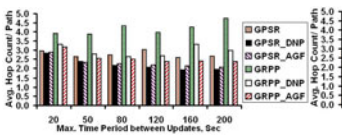


Fig. 3.1. $v_{max} = 5$ m/s

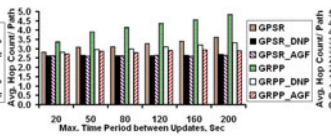


Fig. 3.2. $v_{max} = 30$ m/s

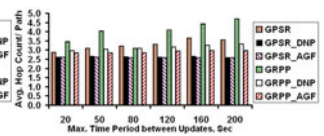


Fig. 3.3. $v_{max} = 50$ m/s

Fig. 3. Average Hop Count per Path (25 Node Network and 15 $s-d$ Pairs)

For a given value of $MaxT_{update}$, the percentage of data packets getting forwarded through perimeter routing for GPSR increases as the maximum node velocity increases. With perimeter routing, due to the lack of nodes lying on a straight line path connecting the source and the destination, packets get forwarded on a path around the straight line. As more data packets get forwarded through perimeter routing, the average hop count of the paths for a session increases. The hop count of GPSR routes increases with node mobility and with increase in the value of $MaxT_{update}$, for a given offered data traffic load.

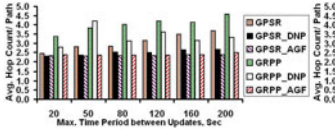


Fig. 4.1. $v_{max} = 5$ m/s

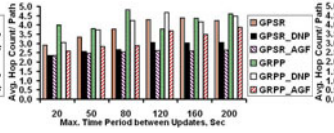


Fig. 4.2. $v_{max} = 30$ m/s

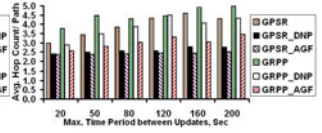


Fig. 4.3. $v_{max} = 50$ m/s

Fig. 4. Average Hop Count per Path (50 Node Network and 30 s - d Pairs)

The hop count of GRPP is higher than that of GPSR by a factor of 20 to 50%. The main reason for this increase in the hop count for GRPP is that while forwarding a data packet towards the destination, the forwarding node chooses the next hop node as the neighbor node that exists (i.e., currently exists and predicted to exist for the next three seconds) closer to the straight line joining the forwarding node's location and the location of the ultimate packet destination, as specified in the data packet header. The hop count would have minimized if the forwarding node chooses the next hop node as the neighbor node that is closer to the destination (similar to the GPSR approach). But, GRPP adopts the “stay on the line through a sequence of stable links” approach to maximize the chances of a data packet reaching the ultimate destination, even if the tradeoff is a higher hop count.

We also observe that for a given offered data traffic load and node mobility, the hop count of GRPP paths in networks of high density is 10 to 25% more than that of the GRPP paths in networks of low density. The advantage is a relatively higher packet delivery ratio (by a factor of 10 to 20%) for GRPP in high-density networks when compared to low-density networks. In high density networks, there are greater chances of finding a source-destination path with all the intermediate nodes located on or closer to the straight line joining the source and the destination. GRPP prefers to go through such “stay on the line” paths (which can have more intermediate nodes) rather than paths that connect the source and destination with the minimum required number of intermediate nodes, but the intermediate nodes are located far away from the straight line joining the source and destination locations.

The hop count of the GRPP routes in networks of moderate and higher node mobility is 15 to 35% more than the GRPP hop count incurred in networks of low node mobility. This could be attributed to the fact that at the time of forwarding a data packet using GRPP, the forwarding node attempts to choose a next hop node as the neighbor node that would be connected to it for the immediate future so that the data packet sent to the neighbor node does not get lost due to the neighbor node suddenly moving away. In order for a forwarding node to choose a neighbor node with which a stable link is predicted to exist at least for the immediate future, the forwarding node and the neighbor node should be either moving towards each other or be moving parallel to each other separated by a smaller distance.

The DNP and AGF versions are observed to be very effective in reducing the hop count per path incurred by the original versions of both GPSR and GRPP. Both DNP and AGF are effective strategies to avoid routing loops encountered with the original versions of GPSR and GRPP. If the destination node is located in the one-hop neighborhood (in the case of DNP and AGF) or is located in the two-hop neighborhood (in the case of AGF), a data packet is forwarded to the destination node (if located in the

one-hop neighborhood) or to the neighbor of the destination node (if located in the two-hop neighborhood) and is not forwarded based on the destination location co-ordinates in the data packet header.

3.3 End-to-End Delay per Data Packet

Figures 5 and 6 illustrate the end-to-end delay per data packet for all the routing protocols. GPSR and its improved versions incur a lower end-to-end delay per data packet for all simulation conditions. In networks of low density and high offered traffic load, almost all the nodes play the role of intermediate nodes for at least one $s-d$ session and each node acts as source or destination for at least one session.

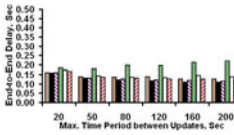


Fig. 5.1. $v_{max} = 5$ m/s

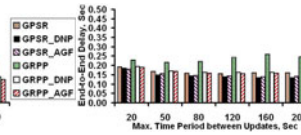


Fig. 5.2. $v_{max} = 30$ m/s

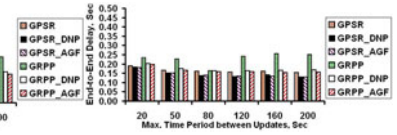


Fig. 5.3. $v_{max} = 50$ m/s

Fig. 5. Average End-to-End Delay per Data Packet (25 Node Network, 15 $s-d$ Pairs)

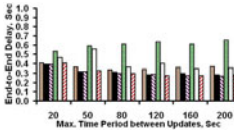


Fig. 6.1. $v_{max} = 5$ m/s

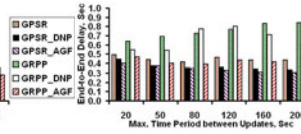


Fig. 6.2. $v_{max} = 30$ m/s

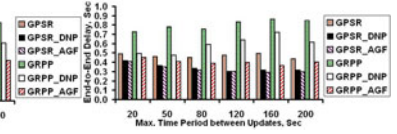


Fig. 6.3. $v_{max} = 50$ m/s

Fig. 6. Average End-to-End Delay per Data Packet (50 Node Network, 30 $s-d$ Pairs)

For position-based routing protocols, the delay per data packet decreases as we use higher values of $MaxT_{update}$. But, the decrease in the delay per data packet is not proportional to the increase in $MaxT_{update}$ value. For a given offered data traffic load and node mobility, the decrease in the end-to-end delay per data packet is by factors of 20-30% (for low density networks) and 10-15% (for high density networks) when the $MaxT_{update}$ value is increased from 20 seconds to 200 seconds. The relative decrease in the magnitude of the difference is attributed to the increase in the route discovery control message overhead (route discoveries and route maintenance/ repair) for LPBR as we increase the offered data traffic load and/or node mobility.

The delay per data packet incurred by GRPP could be as large as 1.5 – 2.0 times to that incurred by GPSR. This could be due to paths of larger hop count chosen by GRPP. The DNP and AGF versions of GRPP reduce the delay significantly by routing the data packet through shortest paths as and when possible. The reduction in the delay per data packet incurred with GRPP_DNP and GRPP_AGF in comparison with that incurred using GRPP could be as large as by a factor of 1.5 – 2.2.

3.4 Packet Delivery Ratio

Figures 7 and 8 illustrate the packet delivery ratio achieved. The packet delivery ratio of GRPP, GPSR_AGF and GRPP_AGF is the largest in most of the simulation conditions. This is attributed to the “stay on the line through a sequence of stable links” path selection approach of GRPP and the effective one-hop/two-hop neighborhood based destination location identification approach of AGF.

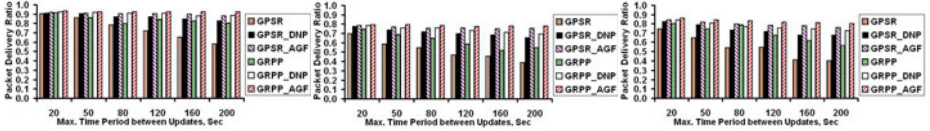


Fig. 7.1. $v_{max} = 5$ m/s

Fig. 7.2. $v_{max} = 30$ m/s

Fig. 7.3. $v_{max} = 50$ m/s

Fig. 7. Average Packet Delivery Ratio (25 Node Network, 15 s - d Pairs)

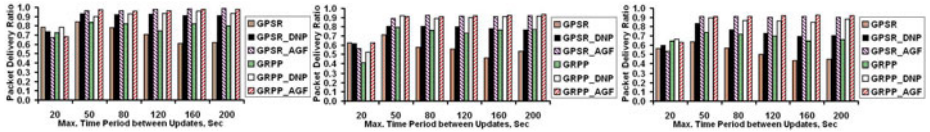


Fig. 8.1. $v_{max} = 5$ m/s

Fig. 8.2. $v_{max} = 30$ m/s

Fig. 8.3. $v_{max} = 50$ m/s

Fig. 8. Average Packet Delivery Ratio (50 Node Network, 30 s - d Pairs)

The packet delivery ratio of GPSR is about 5 to 15% less than that of GRPP. Note that the packet delivery ratios for GPSR and GRPP decrease with increase in the value of $MaxT_{update}$. This is attributed to the lack of accurate information of the destination location as the time period between two successive destination location update searches increases. As a result, data packets are subjected to more of perimeter forwarding and routing loops and hence get dropped eventually. On the other hand, the DNP and AGF versions of both GPSR and GRPP are not much affected by the value of $MaxT_{update}$ because they rely on locally finding the destination node based on the neighborhood information collected during the per-second periodic beacon exchange. This is attributed to the relative reduction in the HLS cellcasting overhead incurred as part of the destination location update search process and availability of more space in the queue of the nodes for forwarding the data packets.

4 Conclusions and Future Work

We compared the performance of GPSR, GRPP protocols and their improved versions GPSR_DNP, GPSR_AGF, GRPP_DNP and GRPP_AGF. We ran extensive simulations (in ns-2) by varying the network density, node mobility and offered data traffic load. We observe a tradeoff between the packet delivery ratio and the destination location search overhead in the case of the position-based routing protocols. As

the maximum time period between two successive destination location searches is reduced, the accuracy of destination location information included in the data packet header is increased leading to an improvement in the packet delivery ratio. But, this is achieved at the cost of a higher destination location search control message overhead. Nevertheless, the DNP and AGF versions of GPSR and GRPP could yield relatively larger packet delivery ratios and still incur a lower control message overhead. The improved versions, GPSR_DNP and GPSR-AGF yield routes with a lower hop count and delay per data packet for most of the scenarios. GRPP and its improved versions yield a relatively higher packet delivery ratio compared to GPSR and its improved versions. As future work, we will evaluate the performance of the position-based routing protocols with other MANET mobility models and also compare them with respect to energy consumption per node and node lifetime.

References

1. Bettstetter, C., Hartenstein, H., Perez-Costa, X.: Stochastic Properties of the random Way Point Mobility Model. *Wireless Networks* 10(5), 555–567 (2004)
2. Bianchi, G.: Performance Analysis of the IEEE 802.11 Distributed Coordination Function. *IEEE Journal of Selected Areas in Communication* 18(3), 535–547 (2000)
3. Broch, J., Maltz, D.A., Johnson, D.B., Hu, Y.C., Jetcheva, J.: A Performance of Comparison of Multi-hop Wireless Ad hoc Network Routing Protocols. In: 4th International Conference on Mobile Computing and Networking, pp. 85–97. ACM, Dallas (1998)
4. Creixell, W., Sezaki, K.: Routing Protocol for Ad hoc Mobile Networks using Mobility Prediction. *International Journal of Ad Hoc and Ubiquitous Computing* 2(3), 149–156 (2007)
5. Fall, K., Varadhan, K.: NS Notes and Documentation, The VINT Project at LBL, Xerox PARC, UCB, and USC/ISI (2001), <http://www.isi.edu/nsnam/ns>
6. Johansson, P., Larsson, N., Hedman, N., Mielczarek, B., Degermark, M.: Scenario-based Performance Analysis of Routing Protocols for Mobile Ad hoc Networks. In: 5th Annual International Conference on Mobile Computing and Networking, pp. 195–206. ACM, Seattle (1999)
7. Karp, B., Kung, H.T.: GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. In: 6th Annual International Conference on Mobile Computing and Networking, pp. 243–254. ACM, Boston (2000)
8. Keiss, W., Fuessler, H., Widmer, J.: Hierarchical Location Service for Mobile Ad hoc Networks. *ACM SIGMOBILE Mobile Computing and Communications Review* 8(4), 47–58 (2004)
9. Naumov, V., Baumann, R., Gross, T.: An Evaluation of Inter-Vehicle Ad hoc Networks based on Realistic Vehicular Traces. In: 7th International Symposium on Mobile Ad hoc Networking and Computing, pp. 108–119. ACM, Florence (2006)
10. Son, D., Helmy, A., Krishnamachari, B.: The Effect of Mobility-Induced Location Errors on Geographic Routing in Mobile Ad hoc Sensor Networks: Analysis and Improvement using Mobility Prediction. *IEEE Transactions on Mobile Computing* 3(3), 233–245 (2004)
11. Yamazaki, K., Sezaki, K.: A Proposal of Geographic Routing Protocols for Location-aware Services. *Electronics and Communications in Japan (Part I: Communications)* 87(4), 26–34 (2003)

Privacy Preserving Naïve Bayes Classification Using Trusted Third Party Computation over Distributed Progressive Databases

Keshavamurthy B.N. and Durga Toshniwal

Department of Electronics & Computer Engineering,
Indian Institute of Technology Roorkee,
Uttarakhand, India
{kesavdec,durgafec}@iitr.ernet.in

Abstract. Privacy-preservation in distributed progressive databases is an active area of research in recent years. In a typical scenario, multiple parties may wish to collaborate to extract interesting global information such as class labels without revealing their respective data to each other. This may be particularly useful in applications such as customer retention, medical research etc. In the proposed work, we aim to develop a global classification model based on the Naïve Bayes classification scheme. The Naïve Bayes classification has been used because of its simplicity, high efficiency. For privacy-preservation of the data, the concept of trusted third party with two offsets has been used. The data is first anonymized at local party end and then the aggregation and global classification is done at the trusted third party. The proposed algorithms address various types of fragmentation schemes such as horizontal, vertical and arbitrary distribution required format. The car-evaluation dataset is used to test the effectiveness of proposed algorithms.

Keywords: privacy preservation, distributed database, progressive database.

1 Introduction

In recent years, due to the advancement of computing and storage technology, digital data can be easily collected. It is very difficult to analyze the entire data manually. Thus a lot of work is going on for mining and analyzing such data.

Of the various techniques of data mining analysis, progressive databases analysis is one of the active areas of research work. Progressive data mining discover the results in a defined period of interest or focus. The Progressive databases have posed new challenges because of the following inherent characteristics such as it should not only add new items to the period of interest but also removes the obsolete items from the period of interest. It is thus a great interest to find results that are up to date in progressive databases.

In many real world applications such as hospitals, retail-shops, design-firms and universities databases, data is distributed across different sources. The distributed database is comprised of horizontal, vertical or arbitrary fragments. In case of horizontal fragmentation, each site has the complete information on a distinct set of entities. An integrated

dataset consists of the union of these datasets. In case of vertical fragments each site has partial information on the same set of entities. An integrated dataset would be produced by joining the data from the sites. Arbitrary fragmentation is a hybrid of previous two.

Distributed progressive databases which constitute the characteristics of both database is distributed either in horizontal, vertical or arbitrarily as well as progressive-ness of the database that is interested in, data within the focus or period of interest.

The key goal for privacy preserving data mining is to allow computation of aggregate statistics over an entire data set without compromising the privacy of private data of the participating data sources. The key methods such as secure multiparty computation use some transformation on the data in order to perform the privacy preservation. One of the methods in distributed computing environment which uses the secure sum multi party computation technique of privacy preservation is Naïve Bayes classification [1].

A few of research papers have discussed the privacy preserving mining across distributed databases. One of important drawback with the existing methods of computation is that the global pattern computation is done at one of the data source itself [2]. This paper addresses this issue effectively by using a trusted third party and it also addresses the privacy preservation naïve bayes classification for Progressive distributed databases.

The rest of the paper is organized as follows: Section 2 briefs about related research work. Section 3 presents privacy preservation Naïve Bayes classification using trusted third party computation over distributed databases. Section 4 gives experimental results. Section 5 includes conclusion.

2 Related Work

Initially, for privacy preserving data mining randomization method were used, the randomization method has been traditionally used in the context of distorting data by probability distribution [3] [4]. In [5] [6], it was discussed how to use the approach for classification. In [5] discusses about privacy protection and knowledge preservation by using the method of anonymization, it anonymizes data by randomly breaking links among attribute values in records by data randomization.. In [6] it was discussed the building block to obtain random forests classification with enhanced prediction performance for classical homomorphic election model, for supporting multi-candidate elections.

A number of other techniques [7] [8] have also been proposed for privacy preservation which works on different classifiers such as in [7], it combine the two strategies of data transform and data hiding to propose a new randomization method, Randomized Response with Partial Hiding (RRPH), for distorting the original data. Then, an effective Naïve Bayes classifier is presented to predict the class labels for unknown samples according to the distorted data by RRPH. In [8], Proposes optimal randomization schemes for privacy preserving density estimation.

The work in [9] [10] describes the methods of improving the effectiveness of classification such as in [9] proposes two algorithms BiBoost and MultBoost which allow two or more participants to construct a boosting classifier without explicitly sharing their data sets and analyze both the computational and the security aspects of the

algorithms. In [10] it proposes a method which eliminates the privacy breach and increase utility of the released database.

In case of distributed environment, the most widely used technique in privacy preservation mining is secure sum computation [11]. Here when there are n data sources $DS_0, DS_1, \dots, DS_{n-1}$ such that each DS_i has a private data item $d_i, i = 0, 1, \dots, n-1$ the parties want to compute $\sum_{i=0}^{n-1} d_i$ privately, without revealing their private data d_i to each other. The following method was presented:

We assume that $\sum_{i=0}^{n-1} d_i$ is in the range $[0, m-1]$ and DS_j is the protocol initiator:

1. At the beginning DS_j chooses a uniform random number r within $[0, m-1]$.
2. Then DS_j sends the sum $d_j + r \pmod{m}$ to the data source $DS_{j+1} \pmod{n}$.
3. Each remaining data sources DS_i do the following: upon receiving a value x the data source DS_i sends the sum $d_i + x \pmod{m}$ to the data source $DS_{i+1} \pmod{n}$.
4. Finally, when party DS_j receives a value from the data source $DS_{n-1} \pmod{n}$, it will be equal to the total sum $r + \sum_{i=0}^{n-1} d_i$. Since r is only known to DS_j it can find the sum $\sum_{i=0}^{n-1} d_i$ and distribute to other parties.

The Naïve Bayes technique applies to learning tasks where each instance x is described by a conjunction of attribute values and the target function $f(x)$ can take on any value from some finite set C . A set of training examples of the target function is provided, and a new instance is presented, described by the tuple of attribute values a_1, a_2, \dots, a_n . The learner is asked to predict the target value, or classification, for this new instance. The Bayesian approach to classifying the new instance is to assign the most probable target value, c_{MAP} , given the attribute values a_1, a_2, \dots, a_n that describe the instance.

$$C_{MAP} = \arg_{\max} P(c_j / a_1, a_2, \dots, a_n) . \tag{1}$$

Using Bayes theorem,

$$\begin{aligned} C_{MAP} &= \arg_{\max} P(a_1, a_2, \dots, a_n / c_j) P(c_j) / P(a_1, a_2, \dots, a_n) . \\ &= \arg_{\max} (P(a_1, a_2, \dots, a_n / c_j) P(c_j)) . \end{aligned} \tag{2}$$

The Naïve Bayes classifier makes the further simplifying assumption that the attribute values are conditionally independent given the target value. Therefore,

$$C_{NB} = \arg \max (P(c_j)) \cap P(a_i/c_j). \tag{3}$$

Where C_{NB} denotes the target value output by the Naïve Bayes classifier.

The conditional probabilities $P(a_i/c_j)$ need to be estimated from the training set. The prior probabilities $P(C_j)$ also need to be decided by some heuristics. Probabilities are computed differently for nominal and numerical attributes.

For a nominal attribute in a horizontally partitioned data, the conditional probability $P(C = c / A = a)$ that an instance belongs to class c given that the instance has an attribute value $A = a$, is given by

$$P(C=c \cap A=a) = \frac{P(C=c \cap A=a)}{P(A=a)} = \frac{n_{ac}}{n_a}. \tag{4}$$

n_{ac} is the number of instances in the (global) training set that have the class value c and an attribute value of a , while n_a is the number of instances in the global training set which simply have an attribute value of a . The necessary parameters are simply the counts of instances. n_{ac} and n_a Due to horizontal partitioning of data, each party has partial information about every attribute. Each party can locally compute the local count of instances. The global count is given by the sum of the local counts. Secure computing a global count is straightforward. Assuming that the total number of instances is public, the required probability can be computed by dividing the appropriate global sums where the local number of instances will not be revealed.

For an attribute with l different attribute values and a total of r distinct classes, $l \cdot r$ different counts need to be computed for each combination of attribute value and class value. For each attribute value a total instance count also to be computed, this gives l additional counts.

3 Progressive Naïve Bayes Classification Using Trusted Third Party Computation over Distributed Databases

3.1 Horizontal Partition

Here each party locally computes the local count of instances. The global count is given by the sum of the local counts. To count global values by summing all local values we use modified secure sum algorithm as shown in the Fig. 1 which is send to trusted third party.

Assumption: n parties, r class values, z attribute values, j^{th} attribute contain l_j different values, S- supervisor, P- parties, C_{lr}^i = No. of instances with party P_i having classes r and attribute values l, N_r^i = No. of instances with party P_i having classes r.

At P:

Algorithm getData(POI)

```
{
While (new data is arriving on the site)
    For (each n parties having specific attributes)
        updateValues( $c_{yz}^i, n_y^i$ );
Encrypt the values and send to trusted third party;
}
```

At S:

Receive values and decrypt them;

From all parties, we can get all possible C_{lr}^1 and N_r^1

Parties calculate the required probabilities from C_{lr}^1 and N_r^1 , on that basis will predict the class.

Fig. 1. Algorithm for Horizontal Scenario

The Update Value function of horizontal scenario shown in Fig.2 will be described as follows: As in the progressive database, the new data is keep on arriving at every timestamp and data become obsolete to keep database up to data, so we also have to n_{ac} and n_a at every timestamp. In the algorithm, we keep on updating the n_{ac} and n_a until $t_{current}$ is less than POI, as no data become obsolete so we keep adding number of instances in both n_{ac} and n_a .

As $t_{current}$ exceeds POI then we also have to remove the instances which are no more in the required period of interest. At every timestamp, we store the records which will be obsolete in next time stamp, as these records have the values which have to be reduced to update n_{ac} and n_a . As the time increases new data will come and also we have the n_{ac} of obsolete data. We update the n_{ac} and n_a by adding the values of n_{ac} and n_a of new data, and by removing values of n_{ac} and n_a by the obsolete data as shown in updateValue function.

```

void UpdateValues (List  $c_{yz}^i$ , List  $n_y^i$ )
{
  For transaction at  $t = t_{current}$ 
  {
    For all class values  $y$  do
    For all  $z$ , Party  $P_i$  locally computes  $c_{yz}^i$ 
    Party  $P_i$  locally computes  $n_y^i$ 
    End For
  }
   $c_{yz}^i$  (new) =  $c_{yz}^i$  (previous) +  $c_{yz}^i$  (transaction at  $t = t_{current}$ )
   $n_y^i$  (new) =  $n_y^i$  (previous) +  $n_y^i$  (transaction at  $t = t_{current}$ )
  If ( $(t_{current} - POI) > 0$ )
  { For transaction at  $t = (t_{current} - POI)$ 
    { For all class values  $y$  do
      For all  $z$ , Party  $P_i$  locally computes  $c_{yz}^i$ 
      Party  $P_i$  locally computes  $n_y^i$ 
      End For }
     $c_{yz}^i$  (new) =  $c_{yz}^i$  (current) -  $c_{yz}^i$  (transaction at  $t = t_{current} - POI$ )
     $n_y^i$  (new) =  $n_y^i$  (current) -  $n_y^i$  (transaction at  $t = t_{current}$ )
  } }

```

Fig. 2. Update value function

3.2 Vertical Partition

In nominal attributes, each party calculates their local instances of n_{ac} and n_a , of the attribute values they have. As each party have different attributes, so no parties have same value of instance of attribute and class. Hence there is no need to calculate the sum of values. At a particulate timestamp, we calculate the local values of n_{ac} and n_a , and send them to the trusted third party The necessary algorithm is given above in Fig.3.

Assumption: n parties, r class values, z attribute values, j^{th} attribute contain l_j different values,
 S – supervisor, P - parties, C_{lr}^i = No. of instances with party P_i having classes r and attribute
 values l , N_r^i = No. of instances with party P_i having classes r .

At P :

Algorithm getData(POI)

{

While (new data is arriving on the site)

 For (each n parties having specific attributes)

 updateValues(c_{yz}^i, n_y^i);

 Encrypt the values and send to trusted third party;

}

At S :

Receive values and decrypt them;

From all parties, we can get all possible C_{lr}^1 and N_r^1

Parties calculate the required probabilities from C_{lr}^1 and N_r^1 , on that basis will predict the class.

Fig. 3. Algorithm for vertical fragmentation

3.3 Arbitrary Partition

In this fragmentation scenario, we consider the mix of both horizontal and vertical fragmentation. Some of the parties have their database distributed horizontally and some having the horizontal part distributed vertically. The trusted third party knows about the partition of different parties with all attributes and class values. The method proposed here uses the ideas discussed in the previous two sections. In this case, we apply our new modified scheme of secure multiparty computation.

4 Results

The dataset used for the purpose of experimentation is car-evaluation [12]. The analysis results of different partitions of distributed progressive databases at a specific period of interest are given as follows and the classification accuracy is in percentage.

Table 1. Horizontal partition classification analysis

Sl. No.	Description	Number of parties	Total number of records	%Accuracy
1	Classification of data at single party (No distribution)	1	1728	85
2	Classification of data Distributed in horizontal partition with trusted third party	3	Party1: 500 Records Party2: 500 Records Party3: 728 Records	85

Table 2. Vertical partition classification analysis

Sl. No.	Description	Attributes per parties	Total number of records	%Accuracy
1	Classification of data at single party (No distribution)	7	1728	85
2	Classification of data distributed in vertical partition with trusted third party	Party1: 2 attributes Party2: 2 attributes Party3: 3 attributes	1728	85

Table 3. Arbitrary partition classification analysis

Sl. No.	Description	Total number of parties	Total number of records/attributes per parties	Total number of records	% Accuracy
1	Classification of data Distributed in horizontal partition with trusted third party	2	Party1: 500 records Party2: 500 records	1000	85
2	Classification of data distributed in vertical partition with trusted third party	3	Party1: 2 attributes Party2: 3 attributes Party3: 2 attributes	728	85
3	Classification of data distributed in arbitrary partition with trusted third party (combination of horizontal and vertical partition)	5	7 attributes	1728	85

5 Conclusion

In our proposed work, we have proposed a set of algorithms for classifying data using Naïve Bayes from a group of collaborative parties without breaching privacy. The Naïve Bayes approach is used because of its simplicity and high efficiency. The non-distribution and various distribution scenarios such as horizontal, vertical and arbitrary scenarios are compared and their accuracy is calculated on the data. The accuracy comes out to be the same showing that the algorithm is giving best case results. Privacy is also preserved using privacy preservation techniques such as offset computation and encryption. The third party concept is also introduced to calculate global classification results with privacy preservation. In this case, Naïve Bayes algorithm is applied to the different distributed progressive sequential data streams scenarios such as horizontal, vertical and arbitrary. In future algorithm can be modified for numeric data to widen its scope.

References

1. Vaidya, J., Kantarcioğlu, M., Clifton, C.: Privacy-Preserving Naïve Bayes classification. *International Journal on Very Large Data Bases* 17(4), 879–898 (2008)
2. Huang, J.-W., Tseng, C.-Y.: A General Model for Sequential Pattern Mining with a Progressive Databases. *IEEE Trans. Knowledge Engineering* 20(9), 1153–1167 (2008)
3. Liew, C.K., Choi, U.J., Liew, C.J.: A data distortion by probability distribution. *ACM TODS*, 395–411 (1985)
4. Warner, S.L.: Randomized Response: A survey technique for eliminating evasive answer bias. *Journal of American Statistical Association* (60), 63–69 (1965)
5. Agarwal, R., Srikanth, R.: Privacy-preserving data mining. In: *Proceedings of the ACM SIGMOD conference*, vol. 29, pp. 439–450 (2005)
6. Agarwal, D., Agarwal, C.C.: On the design and Quantification of Privacy-Preserving Data Mining Algorithms. In: *ACM PODS Conference*, pp. 1224–1236 (2002)
7. Zhang, P., Tong, Y., Tang, D.: Privacy-Preserving Naïve Bayes Classifier. In: Li, X., Wang, S., Dong, Z.Y. (eds.) *ADMA 2005. LNCS (LNAD)*, vol. 3584, pp. 744–752. Springer, Heidelberg (2005)
8. Zhu, Y., Liu, L.: Optimal Randomization for Privacy-Preserving Data Mining. In: *KDD ACM KDD Conference* (2004)
9. Gambs, S., Kegl, B., Aimeur, E.: Privacy-Preserving Boosting. *Journal* (to appear)
10. Poovammal, E., Poonavaikko, M.: An Improved Method for Privacy Preserving Data Mining. In: *IEEE IACC Conference*, Patiala, India, pp. 1453–1458 (2009)
11. Yao, A.C.: Protocol for secure sum computations. In: *Proc. IEEE Foundations of Computer Science*, pp. 160–164 (1982)
12. Bohanec, M., Zupan, B.: *UCI Machine Learning Repository*. (1997), <http://archive.ics.uci.edu/ml/datasets/Car+Evaluation>

Cluster Based Mobility Considered Routing Protocol for Mobile Ad Hoc Network

Soumyabrata Saha¹ and Rituparna Chaki²

¹ Department of Information Technology,
JIS College of Engineering, West Bengal, India
som.brata@gmail.com

² Department of Computer Science & Engineering,
West Bengal University of Technology, West Bengal, India
ritu.chaki@gmail.com

Abstract. Mobile ad hoc networks consist of wireless hosts that communicate with each other without the support of fixed infrastructure. In case of large network, a flat structure may not be the most efficient organization for routing. For this purpose many clustering schemes have been proposed that organize the MANET into a hierarchy, with a view to improve the efficiency of routing. In this paper, we have presented a brief review of the state of the art scenario of routing topologies for mobile ad hoc networks and try to present a scheme that leads to cluster formation which efficiently uses the resources of the MANET. The proposed Cluster Based Mobility Considered Routing Protocol obtains efficient communications among MANET and achieves scalability in large networks by using the clustering technique. The new algorithm takes into consideration the mobility factor during routing and as well as computational overhead is also diminished.

Keywords: Mobile ad hoc networks, routing, clustering, mobility, MANET.

1 Introduction

Mobile ad hoc network is an autonomous system of mobile nodes connected by wireless links; each node operates as an end system for all other nodes in the network. But there are some constraints such as low bandwidth, limited energy, mobility, non-deterministic topology and the broadcast nature of wireless communication make the task complex. Wireless nodes that communicate with each other, forms a multi hop packet radio network and maintains connectivity in a decentralized manner.

Existing research work in ad hoc network's protocols mostly aims at improving stability of routes and creating a collaborative environment between nodes. In the absence of prior knowledge, as in the case of mobile ad hoc network, the work becomes more difficult. Jie Wu and Fei Dai [9] say that mobility of nodes adds another dimension of complexity in the mutual interference. T.Camp, J.Boleng and V.Davies gave an excellent survey on mobility models for MANETs [12]. The popular mobility models include (1) random walk, which is a simple mobility model based on random directions and speeds; (2) random waypoint, which includes pause time between

changes in destination and speed; (3) random direction mobility, which forces hosts to travel to the edge of the simulation area before changing direction & speed.

Clustering has evolved as an important research topic in Mobile ad hoc networks, and it improves the system performance of large MANETs. Clustered based ad hoc networks can distribute resources in a balanced way and are with special advantages related to scalability, efficient energy consumption, and simple data aggregation. To adopt clustering scheme, a mobile ad hoc network is usually organized into a hierarchical structure of multiple virtual subnets, which form a high-level and relatively stable backbone network. Clustering is a technique to dynamically group nodes in a network into logically separating or overlapping entities called clusters. The clustering structure divides a network into one or several clusters, one of which consists of one cluster head and some cluster members. However, in a clustering network the cluster head serves as a local coordinator for its cluster, performing inter-cluster routing, data forwarding and has to undertake heavier tasks so that it might be the key point of the network. Thus reasonable, cluster head election is important to the performance of the mobile ad hoc network. A cluster gateway is a non cluster-head node with inter-cluster links, so it can access neighboring clusters and forward information between clusters.

The rest of the paper is organized in the following way. In section 2, a comparative study of some of the existing routing topologies has been carried out. We have design and describe the new routing protocol to reduce the overhead for maintaining all routing information for each mobile node, in the section no 3. This will eventually reduce the overhead of maintaining a large database and also save large amount of network capacity, which is required for maintaining current information. Intensive performance evaluations are presented in section 4. Finally in section 5, a conclusion has been summarized.

2 Related Works

In recent years clustering is a very well known routing technique for mobile ad hoc networks. Different clustering algorithms have different optimizations, such as minimum cluster head election and maintenance overhead, maximum cluster stability, maximum node lifespan, etc. In this section we have studied many routing protocols which have been proposed for mobile ad hoc networks.

The Cluster-based Inter-domain Routing [4] protocol obtains efficient communications among MANETs. CIDR [4] protocol has ability to control the overhead reduction. The Cluster-based Inter-domain Routing [4] protocol achieves scalability in large networks by using the clustering technique. This approach exploits the clustering by group affinity. In each domain, the distributed clustering algorithm discovers the set of “traveling companions” and elects within each set a Cluster Head for each affinity group. In CIDR [4], packets to remote nodes are routed via cluster-head advertised routes and packets to local destinations are routed using the local routing algorithm. In this CIDR [4] protocol the cluster head in the subnet acts as local DNS for own cluster and also for neighbor clusters. The cluster head advertises to neighbors and the rest of the network its cluster information. This algorithm has an ability of dynamic discovery of route and dynamic split/merge of route. It uses FSR

[11] like technique for controlling overhead reduction. For enhancing its scalability it implements member digest with Bloom Filter. For evolving membership in time, CIDR [4] preserves its legacy routing scheme in each MANET. So, it is scalable in size and robust to mobility.

Cluster-head Gateway Switch Routing [17] is a hierarchical routing protocol where the nodes are grouped into cluster. In CGSR [17], there is no need to maintain a cluster hierarchy. Instead, each cluster is maintained with a cluster-head, which is a mobile node elected to manage all the other nodes within the cluster. This node controls the transmission medium and all inter-cluster communications occur through this node. The advantage of CGSR [17] protocol is that each node maintains routes to its cluster-head which means that routing overheads are lower compared to flooding routing information through the entire network. However, there are significant overhead with maintaining clusters. This is because each node needs to periodically broadcast its cluster member table and updates its table based on the received updates.

Cluster Based Routing Protocol [14] is a cluster on-demand source routing protocol. It has many similarities with the Dynamic Source Routing Protocol [10]. In CBRP [14], cluster head manages all cluster numbers all the information and behavior in each cluster, and finds the adjacent clusters for routing through the gateway node. By clustering nodes into groups, the protocol efficiently minimizes the flooding traffic during route discovery and speeds up this process as well. Its route shortening and local repair features make use of the 2-hop-topology information maintained by each node through the broadcasting of HELLO messages. CBRP [14] has small routing control overhead, less network congestion and search time during routing.

The Border Gateway [7], [18] protocol is the de-facto inter-domain routing protocol for the Internet. BGP [7], [18] provides a standard mechanism for inter-domain routing among heterogeneous domains. The principle of BGP [7], [18] is to enable opaque interoperation, where each domain has the administrative control over its intra-domain routing protocol and inter-domain routing policy. In Border Gateway Protocol [7], [18] the routes to an internal destination within the same domain are determined by an intra-domain routing protocol, where as the routes to an external destination are determined by the inter domain routing policies among domains. BGP [7], [18] relies on a path vector protocol for exchanging inter-domain level reachability information.

In Hybrid Cluster Routing [8], nodes are organized into a hierarchical structure of multi-hop clusters using a stable distributed clustering algorithm. Each cluster is composed of a cluster head, several gateway nodes, and other ordinary nodes. The cluster head is responsible for maintaining local membership and global topology information. In HCR [8], the acquisition of intra-cluster routing information operates in an on demand fashion and the maintenance of inter-cluster routing information acts in a proactive way. The aim of HCR [8] is to acquire a better balance between routing overhead and latency delay than previous protocols. In HCR [8], the high-level routing information is maintained via a proactive method while the low-level routing information is acquired via an on-demand method.

K-Hop Cluster Based Routing Protocol [5] enlarges the range of electing cluster head to K-hops and introduces the concept of metric of constraint degree to restrain the cluster head election. KHCBRP [5] also improves the routing discovery by integrating the inter-cluster on-demand routing and the intra-cluster table-driven routing.

In KHCBRP [5], the neighbor list is amended to introduce a new data structure, k-hop neighbor table. The benefit is that nodes can directly get information of K-hop neighbors from there k-hop neighbor table in the cluster head election.

Low-Energy Adaptive Clustering Hierarchy [13] algorithm is one of the most popular cluster-based routing protocols for sensor network. LEACH [13] organizes all sensor nodes in the network into several clusters and elects one cluster head for each cluster. The cluster head collects data from the member nodes in the cluster and aggregates the data before forwarding them to the distant base station. The performance of clustering algorithm is influenced by the cluster head election method. The operation of LEACH [13] is separated into two phases, the setup phase and the steady state phase. In the setup phase, the clusters are organized and cluster heads are elected. In the steady state phase, the actual data will be transmitted to the base station. LEACH [13] randomly selects a few sensor nodes as cluster heads and rotates this role to evenly distribute the energy load among sensors in the network. LEACH [13] is able to increase the network lifetime, the nodes far from BS will die fast because cluster heads located away from the BS consume more energy under the single hop communication mode.

Two Step Cluster Head Selection [6] routing protocol, is used to solve the cluster head number variability problem of LEACH [13]. Two Step Cluster Head Selection [6] routing is to prolong the lifetime of a sensor network. In TSCHS [6], cluster head selection is divided into two stages. Firstly, temporary cluster heads are selected in initial selection stage with the number larger than the optimal value and then cluster heads of optimal number are chosen out of the temporary cluster heads according to both residual energy and distances from them to the base station. TSCHS [6] can balance the energy load significantly and increase the lifetime of the network.

Cluster Head Load Balanced Clustering [3] routing protocol is used to load balanced among cluster heads for wireless sensor networks. CHLBC [3] builds backbone network's inter-cluster transit route which is composed of cluster heads and calculates relay traffic of cluster heads. CHLBC [3] elects uneven distributed cluster heads according to the distance from sensors to the base station. CHLBC [3] builds up the hop number field from cluster heads to the base station and constructs inter-cluster transit routes in the backbone network which is composed of cluster heads. Cluster heads calculate relay traffic and sensors select corresponding cluster head according to not only the distance with cluster heads but also the relay traffic of cluster heads. The cluster head with heavy relay traffic has less cluster members, so the load of cluster heads will be balanced.

In the above discussion we have discussed about various existing cluster based routing protocols. In our proposed routing protocol we have considered two important issues of MANETs. One is the clustering technique and other is the node mobility. For consider this purpose, some mobility related algorithms have been also discussed.

In DREAM [16] each mobile node (MN) maintains a location table for all other nodes in the ad hoc network. A location packet from each MN to a nearby MN is transmitted in the ad hoc network at a given frequency and too far away MNs at another lower frequency. In this way by making a difference between far away and nearby nodes, A Distance Effect Algorithm for Mobility [16] attempts to reduce the overhead of location packets. Each location packet contains the coordinate of the source node, the source node's speed, and the time of transmission of that LP. DREAM [16] defines a

timeout value on location information, i.e. when the limit specified for location information exceeds then source resorts to the recovery procedure.

An Advanced Mobility Based Ad Hoc Routing Protocol for Mobile Ad Hoc Networks [1] is based on mobility control to address connectivity, link availability and consistency issues. According to this algorithm mobility of node k is defined as,

$$\gamma_k = \frac{\sum_{i=1}^N (loct_j - loct_i) / (t_j - t_i)}{1} \quad (1)$$

Where, $loct_j$ and $loct_i$ gives the position of the node at time t_j and t_i respectively. Here every node maintains only its neighbor's information. After the calculation of node mobility, it only considers least mobile nodes. For next hop selection it also takes less mobility as a criterion, i.e., it selects least mobile node as next hop. The main objective of AMOBIROUTE [1] is reduce overhead of maintaining routing information. For this purpose this algorithm maintains information only next neighbor and maintains information about lesser mobile nodes.

In Mobile Ad hoc Network Routing Protocol [2] we can calculate the displacement of mobile nodes with respect to time, by using beacon messages. In MOADRP [2], we have also considered the angle of displacement. The main objective of MOADRP [2] is to reduce the overhead of maintaining routing information and as well as to reduce the time delay for finding the route.

In the next section we are going to propose a new routing protocol and try to reduce the problems of previously discussed routing protocols.

3 Proposed New Routing Protocol

The basic advantage of hierarchical cluster routing is that it generally produces less routing overhead than flat routing protocol in mobile ad hoc networks. The hierarchical structures are more flexible routing schemes, which may in turn help well in large scale MANETs. In mobile ad hoc network, one of the important challenges of routing is its node mobility. High mobile nodes very frequently change their position. For maintaining information about those highly mobile nodes, we have to update routing table very frequently. For this reason, in the proposed routing protocol, i.e. in Cluster Based Mobility Controlled Routing Protocol or CBMCRP, we have characterized mobile nodes according to their mobility.

In this paper, we use the self-organizing principles for binding a node to a cluster and as well as to minimize the explicit message passing in cluster formation. We also used the beacon message for keeping track of nodes in cluster. Thus, there is no need for explicit message passing during cluster maintenance.

In this routing protocol, we have designed three clusters of nodes. At first we have to know all nodes' mobility. To find out the mobility of each mobile node we follow the method described in MOADRP [2]. Then according to node mobility, nodes are placed in the appropriate cluster. Each cluster must have a predefined cluster head. The algorithm is basically divided in three parts. The first part involves in the cluster formation. In addition, we choose the cluster gateway during cluster formation avoiding the need to explicitly discover the gateways, thus reducing further the transmission overhead. Every

cluster head maintains information of all the nodes within that cluster. For doing this, cluster head maintains a $node_{infolist}$. The $node_{infolist}$ maintains the current position of the cluster nodes and its recent mobility.

Table 1. Node Information: $Node_{infolist}$

node_{id}	node_{posn}	node_{mob}	time

The second part is used to cluster head selection. In this algorithm a cluster head acts only as an identifying label for cluster entity. And the last part is designed of finding a path to the desired destination.

Table 2. Data Dictionary

Variable	Description
mob_{mid}	A middle range of mobility.
mob_{high}	A high range of mobility.
clr_{low}	Cluster designed for low mobility range.
clr_{mid}	Cluster designed for mid mobility range.
clr_{high}	Cluster designed for high mobility range.
$node_{infolist}$	Current position of the cluster nodes & its recent mobility.
$node_{id}$	Id of a mobile node.
$node_{posn}$	Position of mobile nodes.
$node_{mob}$	Mobility of mobile nodes.
S_n	Source Node.

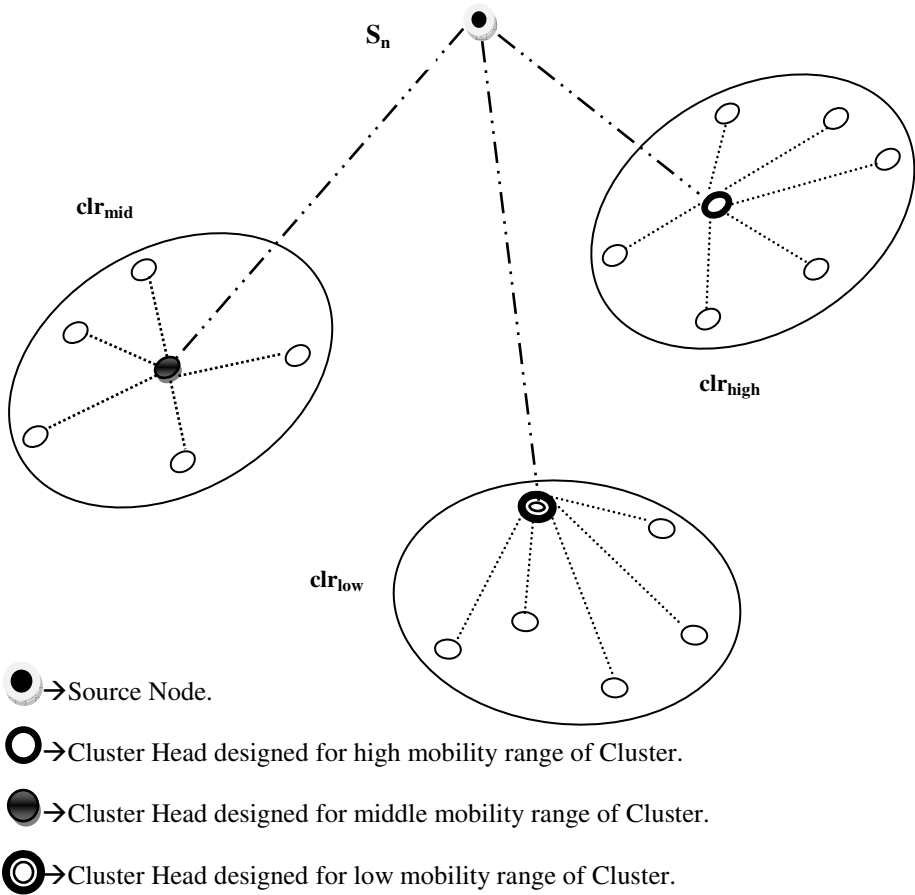


Fig. 1. Formation of Clusters According to CBMCRP

▪ **Cluster Formation:**

Step1: START.

Step2: Source node first calculates mobility of each node.

Step3: If the mobility is less than Mob_{mid}

Then it placed in Clr_{low} cluster.

Else if the mobility is greater than Mob_{mid} and less than Mob_{high}

Then it placed in Clr_{mid} cluster

Else it is placed in Clr_{high} cluster.

Step4: END.

▪ **Cluster Head Selection:**

Step1: START.

Step2: Sort all mobile nodes in each cluster with respect of their mobility.

Step3: The least mobile node selected as cluster head.

Step4: END.

▪ **Routing Algorithm:**

Step1: START.

Step2: Search destination node in Clr_{low}'s cluster head's node_{infolist}

Step i: If destination node is available

Then go to Step 5.

Else go to step 3.

Step3: Search destination node in Clus_{mid}'s cluster head's node_{infolist}

Step j: If destination node is available

Then go to Step 5.

Else go to step 4.

Step4: Take information of destination node from Clus_{high}'s cluster head's node_{infolist}.

Step5: Send destination node information to source node.

Step6: END.

By using the above discussed algorithm a node can communicate with any another nodes within that network.

4 Performance Analysis

The simulation model consists of a network model that has a number of mobile wireless node models, which represents the entire network to be simulated. The number of the nodes ranges from (3-25) nodes depending on the simulation scenario.

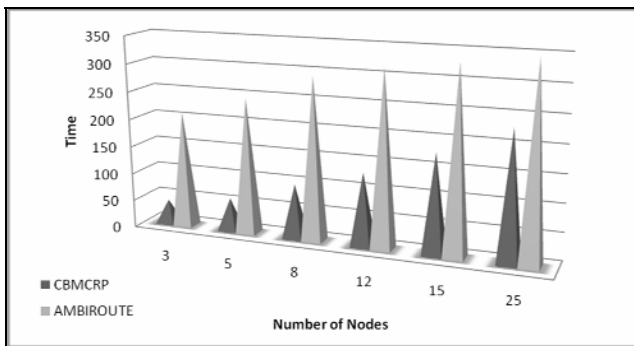


Fig. 2. Number of Nodes vs. Time Graph

The graph in fig 2, shows the time required for finding route from source to destination for a different number of mobile nodes. From that it is clearly seen that when the number of nodes in the network are increased then the required time for route finding also increased.

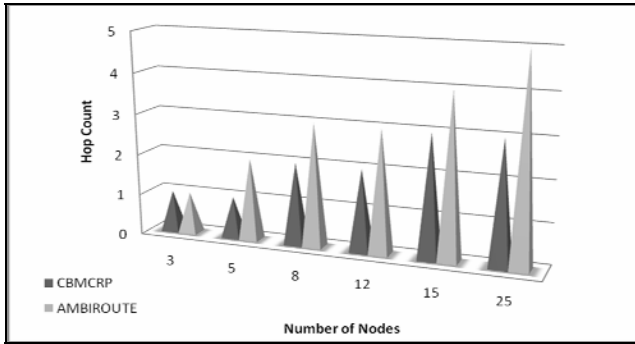


Fig. 3. Number of Node vs. Hop Count Graph

In fig. 3, there is a comparison between number of nodes and hop-count for finding route from source to destination. Here we have also observed that if the number of nodes increase, initially hop count also increase. But after a certain value, hop count fixed in a certain range.

From the above two figures, we have observed that our new proposed and implemented routing algorithm CBMCRP performs better and its time complexity is also less than other routing protocol named AMOBIROUTE [1].

5 Conclusions

In this paper, we have summarized the generic characteristics of well known cluster based routing protocols and present a new routing protocol named Cluster Based Mobility Considered Routing Protocol. This newly proposed algorithm, CBMCRP is based on load balancing approach among cluster heads for mobile ad hoc networks. Mobility is the basic characteristics of mobile node and as well as it is the main challenge for routing in MANETs. For this reason, node mobility is taken one of the important characteristics of this clustering algorithm. To design the different cluster, we have considered different node mobility and according to node mobility, nodes are placed in the appropriate cluster. The cluster head of each cluster acts as a local co-ordinator for its cluster, performing inter-cluster routing, data forwarding and has to undertake heavier tasks so that it might be the key point of the network. Some result analysis are also incorporated to show, in which way the proposed algorithm work to achieve the scalability, the robustness to mobility in large network and some performance analysis have been done with the existing routing algorithm for MANET.

References

1. DasGupta, S., Chaki, R.: AMOBIROUTE: An Advanced Mobility Based Ad Hoc Routing Protocol for Mobile Ad Hoc Networks. In: IEEE International Conference on Networks & Communications, NetCoM 2009 (2009)

2. Saha, S., DasGupta, S., Chaki, R.: MOADRP: Mobile Ad hoc Network Routing Protocol. In: 5th IEEE International Conference on Wireless Communication and Sensor Networks, WCSN 2009 (2009)
3. Jiang, H., Qian, J., Zhao, J.: Cluster Head Load Balanced Clustering Routing Protocol for Wireless Sensor Networks. In: Proceedings of the IEEE International Conference on Mechatronics and Automation, Changchun, China, August 9 - 12 (2009)
4. Zhou, B., Caoan, Z., Gerla, M.: "Cluster-based Inter-domain Routing (CIDR) Protocol for MANETs. In: Sixth IEEE International Conference on Wireless On-Demand Network Systems and Services, WONS 2009 (2009)
5. Chunhua, Z., Cheng, T.: A Multi-hop Cluster Based Routing Protocol for MANET. In: 1st International Conference on Information Science & Engineering, ICISE 2009 (2009)
6. Sun, Z.-G., Zheng, Z.-W., Xu, S.-J.: An Efficient Routing Protocol Based on Two Step Cluster Head Selection for Wireless Sensor Networks. In: 5th IEEE International Conference on Wireless Communications, Networking and Mobile Computing, WiCom 2009 (2009)
7. Nicholes, M.O., Mukherjee, B.: A Survey of Security Techniques for the Border Gateway Protocol (BGP). *IEEE Communications Surveys & Tutorials* 11(1), 52–65 (2009)
8. Niu, X., Tao, Z., Wu, G., Huang, C., Cui, L.: Hybrid Cluster Routing: An Efficient Routing Protocol for Mobile Ad Hoc Networks. In: The proceedings of IEEE ICC (2006)
9. Wu, J., Dai, F.: A Distributed Formation of a Virtual Backbone in MANETs Using Adjustable Transmission Ranges. In: ICDCS 2004, pp. 372–379 (2004)
10. Johnson, D., Maltz, D., Jetcheva, J.: The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks. In: Internet Draft, draft-ietf-manet-dsr- 07.txt (2002)
11. Gerla, M.: Fisheye State Routing Protocol (FSR) for Ad Hoc Networks. In: Internet Draft, draftietf-manet-fsr-03.txt, (2002) work in progress
12. Camp, T., Boleng, J., Davies, V.: A survey of mobility models for ad hoc network research. *Wireless communication and mobile computing: Special issue on Mobile Ad hoc Networking: Research Trends and Application* 2(5), 483–502 (2002)
13. Heinzelman, W.B., Chandrakasan, A.P., Balakrishnan, H.: Energy-efficient communication protocol for wireless microsensor networks. In: Proceedings of 33rd Hawaii International Conferences on System Sciences (HICSS 2000), pp. 3005–3014 (2000)
14. Jiang, M., Li, J., Tay, Y.C.: Cluster Based Routing Protocol (CBRP) Functional Specification. IETF Internet Draft, draft-ietf-manet-cbrp-spec-01.txt (July 1999)
15. Pei, G., Gerla, M., Hong, X., Chiang, C.C.: A wireless hierarchical routing protocol with group mobility. In: Proceedings of IEEE WCNCM 1999, pp. 1538–1542 (1999)
16. Basagni, S., Chlamtac, I., Syrotivk, V.R., Woodward, B.A.: A Distance Effect Algorithm for Mobility. In: Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom 1998), Dallas, TX, pp. 76–84 (1998)
17. Chiang, C.-C., Wu, H.-K., Liu, W., Gerla, M.: Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel. In: Proceedings of IEEE Singapore International Conference on Networks (SICON), Singapore (April 1997)
18. Rekhter, Y., Li, T.: RFC 1771: a Border Gateway Protocol 4 (BGP-4) (March 1995)

Speech Transaction for Blinds Using Speech-Text-Speech Conversions

Johnny Kanisha¹ and G. Balakrishanan²

¹ Research Scholar, Anna university, Trichirappalli

² Director, Indra Ganesan College of Engineering, Trichirappalli

Abstract. Effective human computer interaction requires speech recognition and voice response. In this paper we present a concatenative Speech-Text-Speech (STS) system and discuss the issues relevant to the development of perfect human-computer interaction. The new STS system allows the visually impaired people to interact with the computer by giving and getting voice commands. Audio samples are collected from the individuals and then transcribed to text. A text file is used, where the meanings for the transcribed texts are stored. In the synthesis phase, the sentences taken from the text file are converted to speech using unit selection synthesis. The proposed method leads to a perfect human-computer interaction

Keywords: STS method, Speech Synthesis, Speech recognition.

1 Introduction

Speech is one of the most vital forms of communication in everyday life. On the contrary the dependence of human computer interaction on written text and images makes the use of computers impossible for visually and physically impaired and illiterate masses. Speech recognition and speech generation together can make a perfect human computer interaction. Automatic speech generation from natural language sentences can overcome these obstacles. In the present era of human computer interaction, the educationally under privileged and the rural communities of any country are being deprived of technologies that pervade the growing interconnected web of computers and communications. Although human computer interaction technology has improved significantly in recent decades, current interaction systems still output simply a stream of words in speech. In speech recognition the unannotated word stream lacks useful information about punctuation and disfluencies that could assist the human readability of speech transcripts [1]. Such information is also crucial to subsequent natural language processing techniques, which typically work on fluent and punctuated input. Speech recognition and speech synthesis as separate phases will not give a good human computer interaction. Recovering structural information in speech and synthesising the words has thus become the goal of many studies in computational speech processing [2], [3], [4], [5], [6], [7], [8]. We describe our approach that comprises advanced method for speech recognition using discrete wavelet transform and unit selection method for speech synthesis [22].

2 Problem Statement

In this paper we explain the concatenation of speech recognition and speech generation which leads to a small dictionary system that can be used by visually impaired people. The proposed approach can be called as STS (Speech-Text-Speech) method of speech transaction where a human can dictate a word ,that can be saved in a text file using STT(Speech to Text)System and the related meaning , searched from another file can be given in speech by the computer using TTS(Text to Speech) method. Speech recognition can be done by using the advanced speech recognition method based upon discrete wavelet transform that has 98% of accuracy.

In this method, first, the recorded signal is preprocessed and denoised with Mels Frequency Cepstral Analysis.The feature extraction is done using discrete wavelet transform (DWT) coefficients; Then these features are fed to Multilayer Perceptron (MLP) network for classification. Finally, after training of neural network ,effective features are selected with UTA algorithm. The speech syththesis can be done by using concatenative speech synthesis which depends upon the unit-selection method of speech synthesis.

3 Architecture of Speech Recognition Method

The architecture of our speech recognition system has been shown in the figure below. Our speech recognition process contains four main stages:

1. Acoustic processing that main task of this unit is filtering of the white noise from speech signals and consists of three parts, Fast Fourier Transform, Mels Scale Bank pass Filtering and Cepstral Analysis.
2. Feature extraction from wavelet transform coefficients.
3. Classification and recognition using backpropagation learning algorithm.
4. Feature selection using UTA algorithm [9].

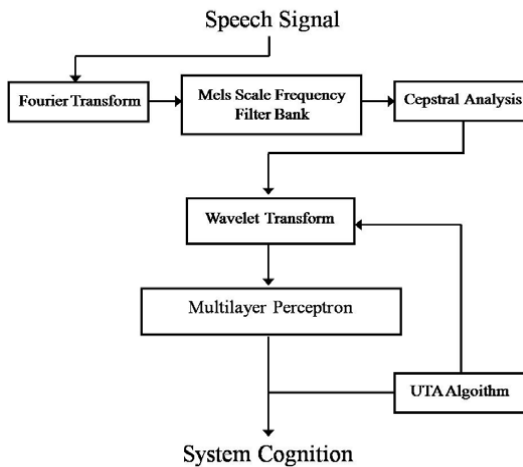


Fig. 1. Speech recognition

4 Unit Selection Synthesis

Unit selection synthesis uses large databases recorded speech[22]. During database creation, each recorded utterance is segmented into some or all of the following: individual phones, syllables, morphemes, words, phrases, and sentences. Typically, the division into segments is done using a specially modified speech recognizer set to a "forced alignment" mode with some manual correction afterward, using visual representations such as the waveform and spectrogram[25]. An index of the units in the speech database is then created based on the segmentation and acoustic parameters like the fundamental frequency (pitch), duration, position in the syllable, and neighbouring phones. Unit selection provides the greatest naturalness, because it applies only small amounts of digital signal processing (DSP) to the recorded speech.

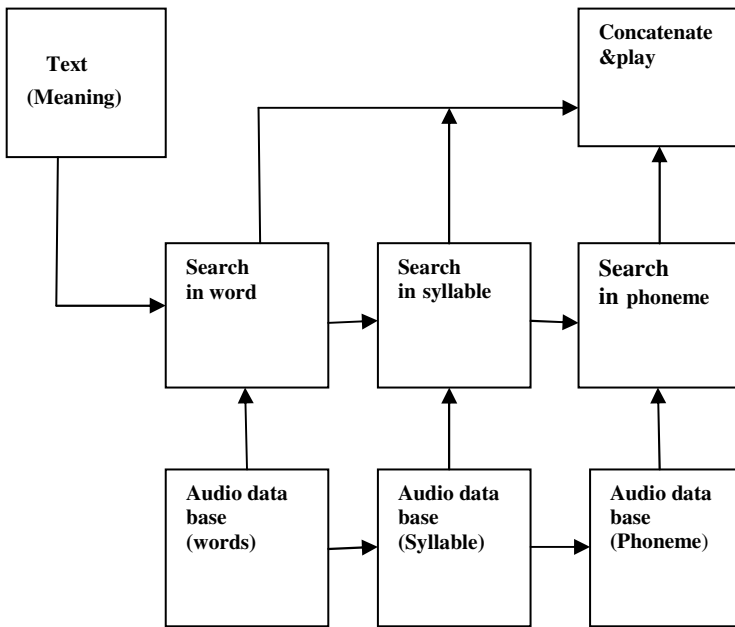


Fig. 2. Speech synthesis

5 Speech-Text-Speech Method

The Speech recognition method using discrete wavelet transform and the speech synthesis method using unit selection process are put together to form STS method. This concatenation is done to create a dictionary application for the blinds to speak a word to the system and to get the meaning for the transcribed word.

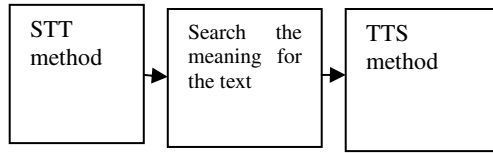


Fig. 3. The STS System

6 Steps for Dictionary System

1. The audio input given by the user should be transcribed to text using the speech recognition method and played back to the user using speech synthesis system

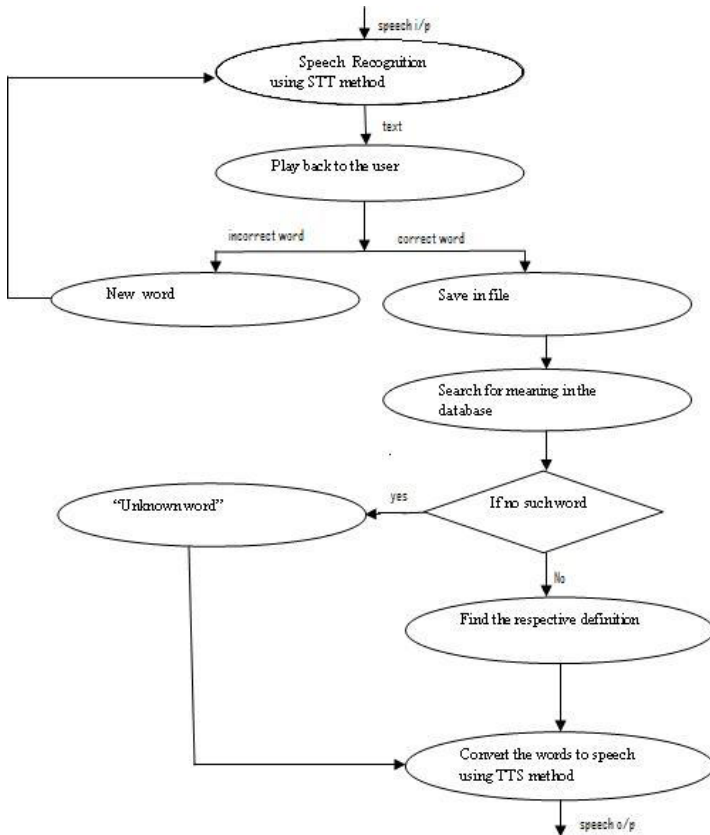


Fig. 4. STS Flow Chart

2. a.If the word is correct and accepted by the user ,the word will be saved in a text file
b.If the word is incorrect then the user has to repronounce it within 10 seconds from the time he heard the word
3. In case of 2.a,the database is interfaced and the word will be checked with the words in the database and the respective meaning defined there will be identified
4. The identified text will be given back to the user in speech through speech synthesis system

7 Conclusion

In this paper, we discussed the issues relevant to the development of perfect human-computer interaction for the benefits of blinds. As man-machine interaction is an appreciated facility even outside the research world,especially for the persons with disabilities, we have designed the STS system,where the blinds can use the system as an audio dictionary.This can be extended to use the web dictionaries.Although being able to give and get speech make the STS system more flexible, the interaction quality may significantly decline unless the recognition and synthesis is done properly.The quality of interaction can be improved if the STT and the TTS system are provided with more descriptive prosody information.

References

- [1] Jones, D., Wolf, F., Gibson, E., Williams, E., Fedorenko, E., Reynolds, D., Zissman, M.: Measuring the readability of automatic speech-to-text transcripts. In: Proc. of Eurospeech, pp. 1585–1588 (2003)
- [2] Heeman, P., Allen, J.: Speech repairs, intonational phrases and discourse markers: Modeling speakers' utterances in spoken dialogue. *Computational Linguistics* 25, 527–571 (1999)
- [3] Kim, J., Woodland, P.C.: The use of prosody in a combined system for punctuation generation and speech recognition. In: Proc. of Eurospeech, pp. 2757–2760 (2001); peech transcripts, In: Proc. of ISCA Workshop: Automatic speech Recognition: Challenges for the Millennium ASR-2000, pp. 228-235 (2000)
- [4] Gotoh, Y., Renals, S.: Sentence boundary detection in broadcast
- [5] Kompe, R.: Prosody in Speech Understanding System. Springer, Heidelberg (1996)
- [6] Snover, M., Dorr, B., Schwartz, R.: A lexically-driven algorithm for disfl uency
- [7] Kim, J.: Automatic detection of sentence boundaries, disfluencies, and conversational fillers in spontaneous speech. Master's thesis, University of Washington (2004)
- [8] Johnson, M., Charniak, E.: A TAG- based noisy channel model of speech repairs. In: Proc. of ACL (2004)
- [9] Meysam, M., Fardad, F.: An advanced method for speech recognition. *World Academy of Science,Engineering and Technology* (2009)
- [10] Kirschning. Continuous Speech Recognition Using the Time-Sliced Paradigm. MEng.Dissertation, University Of Tokushinia (1998)
- [11] Tebelskis, J.: Speech Recognition Using Neural Networks, PhD. Dissertation, School Of ComputerScience, Carnegie Mellon University (1995)

- [12] Tchorz, J., Kollmeier, B.: A Psychoacoustical Model of the Auditory Periphery as Front-end for ASR. In: ASAEAAiDEGA Joint Meeting on Acoustics, Berlin (March 1999)
- [13] Clark, C.L.: Labview Digital Signal Processing and Digital Communications. McGraw-Hill Companies, New York (2005)
- [14] Kehtarnavaz, N., Kim, N.: Digital Signal Processing System-Level Design Using Lab View. University of Texas, Dallas (2005)
- [15] Kantardzic, M.: Data Mining Concepts, Models, Methods, and Algorithms. IEEE, Piscataway (2003)
- [16] Lippmann, R.P.: An Introduction to Computing with neural nets. IEEE ASSP Mag. 4 (1997)
- [17] Martin, H.B.D., Hagan, T., Beale, M.: Neural Network Design. PWS Publishing Company, Boston (1996)
- [18] Dietterich, T.G.: Machine learning for sequential data: A review. In: Caelli, T.M., Amin, A., Duin, R.P.W., Kamel, M.S., de Ridder, D. (eds.) SPR 2002 and SSPR 2002. LNCS, vol. 2396, pp. 15–30. Springer, Heidelberg (2002)
- [19] MathWorks. Neural Network Toolbox User's Guide (2004)
- [20] Kishore, S.P., Black, A.W., Kumar, R., Sangal, R.: Experiments with unit selection Speech Databases for Indian Languages
- [21] Sen, A.: Speech Synthesis in India. IETE Technical Review 24, 343–350 (2007)
- [22] Kishore, S.P., Black, A.W.: Unit size in Unit selection Speech Synthesis. In: Proceedings of Eurospeech, Geneva Switzerland (2003)
- [23] Kishore, S.P., Kumar, R., Sangal, R.: A data – driven synthesis approach for Indian Languages using syllable as basic unit. In: Proceedings of International Conference on National Language Processing, ICON (2002)
- [24] Kawachale, S.P., Chitode, J.S.: An Optimized Soft Cutting Approach to Derive Syllables from Words in Text to Speech Synthesizer. In: Proceedings Signal and Image Processing, p. 534 (2006)
- [25] Segi, H., Takagi, T., Ito, T.: A Concatenative Speech Synthesis Method using Context Dependent Phoneme Sequences with variable length as a Search Units. In: Fifth ISCA Speech Synthesis Workshop, Pittsburgh
- [26] Lewis, E., Tatham, M.: Word and Syllable Concatenation in Text to Speech Synthesis
- [27] Gros, J.Z., Zganec, M.: An Efficient Unit-selection Method for Concatenative Text-to-speech Synthesis

Floating-Point Adder in Technology Driven High-Level Synthesis

M. Joseph¹, Narasimha B. Bhat², and K. Chandra Sekaran³

¹ Mother Teresa College of Engineering and Technology
Illuppur, Pudukkottai - 622102, India

mjoseph_mich@yahoo.com

² Manipal Dot Net Pvt Ltd,
37, Ananth Nagar, Manipal - 576104, India

narasim@manipaldotnet.com

³ National Institute of Technology Karnataka,
Srinivasnagar, Mangalore - 575025, India

kchandrain@yahoo.co.in

Abstract. Implementation of floating-point algorithms in fixed-point processor asks for customization into fixed-point for that processor. Technology driven High-Level Synthesis is a customized High-Level Synthesis approach for a particular fixed-point processor. It makes the present High-Level Synthesis knowledgeable of the target Field Programmable Gate Array. All the functions of High-Level Synthesis become aware of target technology since parsing here. It makes right inference of hardware by attaching target technology specific attributes to the parse tree in it, which guides to generate optimized hardware. This paper, integrating both, presents an approach to synthesize the floating-point algorithms in this customized tool. It performs the conversion of floating-point model into corresponding fixed-point model and synthesizes it for implementing onto an Field Programmable Gate Array. This compiler driven approach generates optimal output in terms of silicon usage and power consumption.

1 Introduction

The cost and power consumption constraints of embedded systems require to use the fixed-point arithmetic for the efficient implementation of Digital Signal Processing (DSP) algorithms. Many DSP and communication algorithms are first simulated using floating-point arithmetic and later transformed into fixed-point arithmetic to reduce implementation complexity.

The efficient implementation of algorithms in hardware architectures like Field Programmable Gate Array (FPGA) requires to minimize the size and the power consumption of the chip. Realization of DSP algorithms with the fixed-point data type provides many benefits, such as savings in power and silicon area.

Methodologies for automatic transformation of floating-point data into fixed-point data are presented in research literature. The transformed fixed-point models will be later synthesized and implemented. We propose a design methodology

to handle the floating-point algorithms in the High-Level Synthesis (HLS) itself. Technology driven HLS (THLS) is a customized HLS approach for a particular target technology. (*Note:Section III presents its details*). We use this THLS framework, to convert the floating-point algorithms into the corresponding fixed-point algorithms, synthesize and generate optimized technology specific netlist for a fixed point processor. This compiler driven approach generates optimized output in terms of silicon usage.

1.1 High-Level Synthesis

HLS takes a source program in any Hardware Description Language (HDL) as input and converts it into Register Transfer Level (RTL) structures. Its front end includes scanner, parser and intermediate code generator. Parser converts the syntax of the HDL input into an annotated parse tree that is then elaborated into an intermediate representation. Elaboration process instantiates modules, evaluates and propagates symbolic constants, checks the connectivity of all the devices and produces a checked consistent design. The Intermediate Representation (IR) is Control/Data Flow Graph (CDFG), a variant of syntax tree along with control information. Its back end consists of optimizer and hardware generator (synthesizer) phases which are scheduling and allocation. The optimizer applies compiler optimization techniques on the CDFG, to improve it, keeping speed, silicon area and power as optimization factors. Scheduling assigns operations to clock cycles. Allocation assigns operations to functional units like arithmetic logic units, multiplexers and storage elements [1], [3], [4]. It uses a generic library of devices like Library of Parameterized Modules (LPM) [10].

1.2 Target Technology

There are two basic versions of Programmable Read Only Memories (PROM) available; one can be programmed by the manufacturer and the other by the end user. The former is Mask Programmable and latter is Field Programmable. The FPGA consists of programmable array of uncommitted elements, which can be interconnected in a generic way [2], [11]. This FPGA is referred as *target technology* in this paper.

The rest of the paper is organized as follows: Section II presents the related work. Section III gives details about the THLS tool. Section IV briefs about floating-point algorithms in THLS. Section V presents the details of an open source floating-point adder and also about THLS compilation details of this adder. Section VI gives the details of the implementation framework. Section VII discusses about the results and implications and section VIII gives the concluding remarks.

2 Related Work

Recently some researchers presented floating-point to fixed-point conversion done by the hardware compilers. Trident is a compiler for floating point

algorithms written in C, producing circuits in reconfigurable logic that exploit the parallelism available in the input description [6].

Sanghamitra Roy et al. presented an approach to automate floating-point to fixed point conversion, for mapping to FPGAs by profiling the expected inputs to estimate errors. Their algorithm attempted to minimize the hardware resources while constraining the quantization error within a specified limit [5].

We propose a technology aware hardware compiler for Verilog HDL to handle floating-point algorithms. This compiler converts the floating-point algorithms into the corresponding fixed-point algorithms, synthesizes and generates optimized technology specific netlist for a fixed point processor.

3 Technology Driven HLS

Present generic HLS approach checks only functional correctness of the design. It neglects the designer's technology knowledge coded into the design since it aims to generate generic output; e.g. it infers an adder for a count operation in the input. It loses the designer's intention of implementing a counter and also does not exploit the counter feature available in the target technology. This will result in sub-optimal implementation. If HLS is aware of target domain knowledge, optimal generation of hardware is possible [4]. To retain designer's technology knowledge coded into the design, and to exploit the target technology to its potential, HLS tool should be made aware of target technology.

THLS is a customized HLS tool for a particular target technology. All the phases of this tool are knowledgeable of the target technology [8]. It infers the hardware rightly, based on the target domain knowledge and generates optimized RTL netlist. In THLS **attributes and target technology** are the key elements. Parser uses AGs to attach target specific attributes, to generate technology specific annotated parse tree. Elaboration then generates *Target Specific Intermediate Representation* (TSIR) from this annotated parse tree. The optimizer then applies compiler optimization techniques on the CDFG, to improve it, keeping speed, silicon area and power as optimization factors. Synthesizer converts this TSIR into hardware. It uses the *Technology Specific Library* (TSL) not any generic library like LPM. Some salient features of this approach are discussed below.

3.1 Right Inference in Parsing

THLS allows the parser to generate annotated parse tree, in which attributes are technology specific. *Attributes are abstractions of target technology features.* These attributes indirectly carry the cost information like speed, silicon area and power. Parser or elaborator use these attributes to infer the operations correctly. This guides to map right hardware devices during synthesis [7]. It also makes all the HLS functions to become aware of target domain.

3.2 TSIR

TSIR is created based on the target technology. Technology details are embedded into the CDFG to make it technology specific. E.g: for shift register, there will be shift register node with the particular technology details.

3.3 TSL

TSL is a library of technology specific devices, defined based on the target technology. E.g: for shift register, there will be shift register device as per target technology.

4 Floating-Point Algorithms in THLS

THLS is a customized HLS synthesis tool for a particular target technology. Implementation of floating-point algorithms in fixed-point processors asks for customization of the those algorithms for that fixed point processor. THLS and floating-point algorithms, in principle, ask for customization for a fixed-point processor. So, THLS will be a suitable framework to handle floating-point algorithms. This methodology exploits the target FPGA for implementing the floating-point algorithms efficiently. This can generate optimized output compared the other conversion (into fixed-point) and compilation methodologies for floating-point algorithms.

5 Floating-Point Adder in THLS

We consider an open source floating point (FP) adder, for our experimentation, designed by Justin Schauer as a summer research project at Harvey Mudd College for Dr. David Harris [9]. We present here the transformation details of this adder in THLS.

5.1 Floating-Point Adder

This adder was designed to fully conform to the IEEE 754 standard. In its fully-featured format, this adder supports all rounding modes: round to nearest even, round to plus infinity, round to minus infinity, and round to zero; it supports denormalized numbers as both input and output; it supports all applicable exception flags: overflow, underflow, inexact, and invalid; it supports trapped overflow and underflow, where the result is returned with an additional bias applied to the exponent. All of these features are supported in hardware. This adder was designed with area minimization in mind, so it is single path, has a leading zero encoder rather than predictor, etc. The adder is also a single-cycle design.

5.2 Floating-Point to Fixed-Point Conversion

Conversion of floating point model into fixed-point model, is performed by a set of Verilog files in the above mentioned adder. These are given as input along with the input floating-point adder program to the hardware compiler. The hardware compiler treats those input conversion files as any other HDL input and synthesizes. Brief outline of the operations involved for the conversion into fixed-point model are given below:

Floating-point alignment. This is an aligner for floating point addition. It takes two single precision floating point numbers and splits out two numbers: the large mantissa with a 1 in the MSB (Most Significant Bit) and the smaller mantissa. Both mantissae also have assumed leading 1's prepended, unless *exponenta* (expa) or *exponentb* (expb) is determined to be 0, in which case no leading 1 is prepended.

Adding mantissae. This block determines the necessary operation to perform on two floating point numbers (addition or subtraction) and then performs that operation. It is a part of a floating point adder, so it takes two properly shifted mantissae with/without leading ones and the signs of the original numbers, and returns the result of the effective operation on the mantissae (the "sum"). It also calculates the *or* of discarded bits for use in rounding.

Normalizing the sum. This block is part of an FP adder. It takes a sum as an input, then shifts the sum so that the MSB is 1. The shift is accomplished by using a priority encoder to encode the position of the leading 1. The sum is then shifted either by this encoded amount, or by $\text{biggerexp} + 1$ (if result is denormal) and the module outputs the amount of the shift, the 23 bit mantissa, and the round and sticky bits. The result is also checked for being zero or denormal, and the *inex* (inexact) flag is set if there are round or sticky bits.

Rounding the sum. This is the rounding module for an FP adder. It takes as inputs the normalized sum, the round bit, the sticky bit, the amount the sum has been shifted so far, the sign of the final result, and the rounding mode, and uses these to determine the appropriate rounding procedure. It also calculates the final exponent, taking into account if there was overflow during rounding.

Piecing together. The final step in this FP adder. It takes the calculated significand, exponent and sign, and using a multitude of flags such as infinities and overflows determines special cases and finally puts together the end floating point result. It also determines all the applicable exceptions: overflow, underflow, invalid, and inexact.

(Note: A set of Verilog files namely *fpalign.v*, *mantadd.v*, *lzencode.v*, *normlize.v*, *rounder.v*, *special.v* and *final.v* are used to perform these operations in this adder.)

5.3 Compiler Transformations

THLS takes Verilog program (*fpadd.v*), as input and converts it into RTL Netlist. Here THLS takes only adder as the input not other supporting conversion input files. THLS synthesizes the floating-point adder after necessary conversion into corresponding fixed-point model. The conversion will be done as part of the compilation process. THLS performs this conversion after parsing and before elaboration.

Consider the Verilog code segment '*out <= out + sig*', which is a floating-point addition operation. Parser in Icarus tool, converts it into a parser expression binary object *PEBinary*. Parser infers an addition operation here. Elaborator then converts this into a structural adder object *NetEBAdd*, a type of *NetEBinary* object. *NetEBAdd* is a node in the generic IR for addition operation. Synthesizer then transforms this into an adder device *NetAddSub* from LPM. This *ivl - LPM* (Icarus Verilog LPM) is a generic object, which can be mapped to any target technology. Optimizer applies optimization techniques and improves this adder. This is the general approach followed in HDL compilers. This is insensitive of the floating-point data. It treats it as a conventional adder not as a floating-point adder.

Parser in THLS, on the other hand, handles it differently. It converts the above code segment into a parser expression binary object *PEBinary*. Parser infers a floating-point addition operation here. Then the conversion of floating-point model into fixed-point model for this adder is performed. All the operations mentioned in the Sub section (Refer:5.2) are performed during elaboration. This technology aware hardware compiler compiles the Verilog HDL code for the floating-point to fixed-point conversion and generates the synthesized netlist. The compiled hardware will actually perform the floating-point to fixed-point conversion.

The compiler transformations for one example operation *adding mantissas* (Refer:5.2) is presented here. Elaboration, after floating-point alignment (Refer: 5.2), generates a structural adder object *NetEBAdd_T*, a type of *NetEBinary* object. *NetEBAdd_T*, here, is a TSIR node. Synthesizer transforms this into an adder device *NetAddSub_T* from TSL. Optimizer then improves it. Sequence of transformations for this operation under THLS tool is given in Figure.1. Here target technology is considered as Virtex IV as an example case. (Note: All objects referred in this paper are C++ objects.)

6 Implementation Framework

We modified Icarus Verilog Compiler to develop the THLS tool. It is a compiler for the IEEE standard 1364 HDL Verilog. It translates the Verilog source code into RTL netlist formats for synthesis or other executable programs for simulation. The currently supported targets are *vvp* for simulation and *xnf* and *fpga* for synthesis. It is an open source EDA (Electronic Design Automation) tool and also a part of gEDA (gnu EDA) [12].

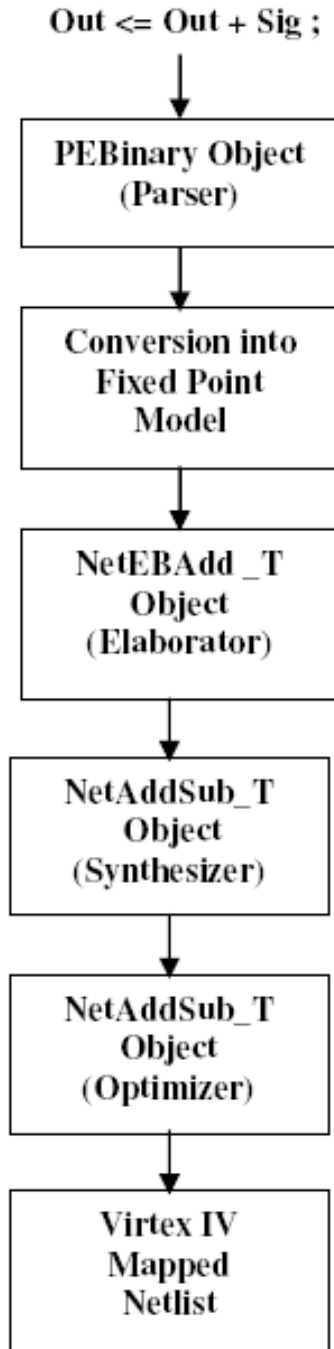


Fig. 1. Compiler transformations for adding mantissas in FPA
FPA - Floating-Point Adder

We used Virtex IV, a high performance FPGA from Xilinx Inc, as the target technology. Its devices are user programmable gate arrays with various configurable elements. The innovative Advanced Silicon Modular Block column based architecture is unique in the programmable logic industry. Virtex-IV contains three families LX, FX and SX, which provide resources to develop logic, communication and digital signal processing applications. A wide array of hard IP-core blocks are also available. Its devices are produced on a state-of-art 90nm copper process using 300mm (12 inch) wafer technology. There is up to 40 speed over previous generation devices in the Configurable Logic Block (CLB) of Virtex-IV [11].

7 Results

We present here the result and analysis of the FPA mentioned in Section 5.1. We used only THLS and ISE synthesis tools for our experimentation of FPA. ISE is leading commercial synthesis tool from Xilinx Inc [11]. The product version used is ISE 8.2.01i. ISE cannot handle floating-point algorithms, but THLS can handle them. ISE compiles the converted fixed-point model for this FPA, whereas THLS converts and compiles them. This comparison is made only to prove that compiler driven approach is better than the conversion and compilation approach for floating-point algorithms. (*Note: Icarus hardware compiler is not able to execute this adder and the respective conversion programs in Verilog. So it is omitted from the comparison*). We executed the above mentioned example FP adder program, in these two synthesis tools and generated synthesized netlists. We could generate programming files successfully in ISE 8.2.01i for the target Virtex IV. We use synthesis and map (technology mapping) reports to compare and analyze the result of these synthesis tools.

We used XPower Estimator 9.1.2 (XPE 9.1.2) for estimating the power consumption by example the circuits considered. XPE 9.1.2 is power estimation and analysis tool from Xilinx Inc,.

7.1 Silicon Usage

Table 1 gives the device utilization details based on map reports generated after technology mapping in ISE, for this adder implementation under ISE and THLS tools. Table 2 gives the details of *total equivalent gate count* for the design based on map reports under these two tools. It is a metric to evaluate the tool's silicon efficiency. It is inferred that THLS tool has **1.36** times improvement in silicon efficiency over ISE i.e. THLS has 26% reduction in silicon usage over ISE. This will lead to optimization in silicon area and in turn power consumption.

7.2 Power Consumption

Table 3 gives the details of power consumption for the FPA implementation under these three synthesis tools. We used the device XC4VLX15 in Virtex-IV

Table 1. Device utilization 1 for FPA

No.	Components	ISE	THLS
1	4 i/p LUTs logic	650	455
2	4 i/p LUTs as route-through	006	009
3	Slices	335	252

(LUT - Look Up Table; FPA - Floating-Point Adder)

Table 2. Device Utilization 2 for FPA

No.	Metric	ISE	THLS
1	Gate Count for design	4560	3348

(Note: Gate Count refers to total equivalent gate count)

Table 3. Power Consumption for FPA

No.	Metric	THLS	ISE
1	Quiescent Power	160 mW	160 mW
2	Dynamic Power	000 mW	001 mW
3	Total Power	160 mW	161 mW

for this experimentation. Total power consumption is 161 mW for ISE tool and 160 mW for THLS tool. THLS is able to reduce the dynamic power 1 mW and thus optimizes power. It has 0.6% reduction in power consumption over ISE. (Note: Power consumption is significant for larger volume applications only. On the other hand, power consumption is less and insignificant in low volume applications.)

Icaurs compiler cannot handle floating-point algorithms. ISE cannot handle floating-point algorithms. THLS customizes the floating-point algorithms into fixed-point and synthesizes. THLS compiler has 26% reduction in silicon usage over the conversion (fixed-point) and compiler methodology (ISE). It has 0.6% reduction in power consumption over ISE.

8 Conclusion

This paper demonstrates that hardware compilers can handle floating-point operations for fixed point processors efficiently. This paper presents an approach for the same and proves that the compiler conversion of floating-point model into fixed point model can produce optimized output in terms of silicon usage.

References

1. McFarland, M.C., Parker, A.C., Campasona, R.: Tutorial on High-Level Synthesis. In: 25th ACM/IEEE Design Automation Conference (1988)
2. Brown, S.D., Francis, R.J., Rose, J., Vranesic, Z.G.: Field Programmable Gate Arrays. Kluwer Academic Publishers, Dordrecht (1992)
3. Gajski, D.D., Dutt, N.D., Wu, A., Lin, S.: High-Level Synthesis: Introduction to Chip and System Design. Kluwer Academic Publishers, Dordrecht (1992); High-Level Synthesis. IEEE Design and Test of Computers (1994)
4. Lin, Y.L.: Recent Developments in High-Level Synthesis. ACM Transactions on Design Automation of Electronic Systems 2(1), 2–21 (1997)
5. Roy, S., Banerjee, P.: An Algorithm for Converting Floating-Point Computations to Fixed-Point in MATLAB based FPGA design. In: Design Automation Conference - DAC 2004, San Diego, California, pp. 484–487 (2004)
6. Tripp, J.L., Peterson, K.D., Ahrens, C., Poznanovic, J.D., Gokhale, M.B.: TRIDENT: An FPGA Compiler Framework for Floating-Point Algorithms. In: International Conference on Field Programmable Logic and Applications, pp. 317–322. IEEE, Los Alamitos (2005)
7. Joseph, M., Bhat, N.B., Chandra Sekaran, K.: Right inference of Hardware in High-Level Synthesis. In: International Conference on Information Processing, ICIP 2007, Bangalore, India (August 2007)
8. Joseph, M., Bhat, N.B., Chandra Sekaran, K.: Technology driven High-Level Synthesis. In: International Conference on Advanced Computing and Communication - ADCOM 2007, IEEE, Indian Institute of Technology Guwahati, India (December 2007)
9. <http://www.hmc.edu/chips>
10. <http://www.edif.org>
11. <http://www.xilinx.com>
12. <http://www.icraus.com>

Differential Artificial Bee Colony for Dynamic Environment

Syed Raziuddin¹, Syed Abdul Sattar¹, Rajya Lakshmi², and Moin Parvez¹

¹ Royal Institute of Technology and Science, Chevella, Hyderabad, India

² Gitam Institute of Technology, Gitam University, Vizag, India
informraziuddin@gmail.com, syed49in@yahoo.com, rdavuluri@yahoo.com

Abstract. This paper introduces a novel variant of artificial bee colony algorithm for complex multimodal and dynamic optimization problem. The Differential Artificial Bee Colony (DABC) is proposed to enhance the bees update strategy for improving the quality of solutions. The DABC is also integrated with external archive to preserve the good solutions produced through the generations and contributing to the better search strategy. Comprehensive analysis of proposed algorithm is carried out on standard benchmark problems with higher dimensions (10, 30 and 50) and on dynamic optimization problems. The algorithmic suitability, robustness and convergence rate are investigated. Results show that the performance of the proposed algorithm is better and competitive to those of the other population based stochastic algorithms.

1 Introduction

Most of the real world science and engineering optimization problems are complex, nonlinear and multimodal functions and a good number of them are dynamic in nature. The need of the increasing complexity of many real-world optimization applications is the efficient and robust optimization algorithms. From the last few decades there has been a growing interest in Swarm Intelligence (SI) algorithms for solving optimization problems. One of the recently introduced SI algorithms is Artificial Bee Colony (ABC) [4]. The ABC is a stochastic, population based optimization algorithm based on intelligence and foraging behavior of bee swarm.

Due to its simplicity and ease of implementation (fewer control parameters), this technique is being used to solve many numerical optimization problems [8,6,9] and real-time optimization problems [10,7,5,1]. In ABC algorithm, the position of the food source represents a possible solution to the optimization problem and the nectar amount corresponds to the quality (fitness) of the associated solution.

Improving the ABC algorithm for quality of solution is an important research area for the ABC community. Though the ABC algorithm is simple to implement with fewer control parameters, it does not sound well on complex multimodal optimization problems with higher dimension and dynamic environment as well. The major pitfalls of the ABC algorithm is the bees update strategy. During the

search process ABC algorithms captures the only best solution being produced for that current iteration and is only used as retention candidate. The onlooker bees and employed bees update their coordinates based on their neighborhood and does not utilize the capability of potential solution being produced. The basic ABC algorithm even lacks the ability to retain a good number of potential solutions that may be produced through the generation during search process.

This paper proposes a novel strategy based on differential update and the external archive, thus named as Differential Artificial Bee Colony (DABC). According to the differential update strategy, the weighted difference of the bee and the neighbor are added to the elite bee. Again an external archive is maintained to retain the number of good solutions produced and the members of the archive are randomly selected for differential update. The archive update strategy is also proposed in this paper. The performance of the proposed algorithm is demonstrated against the present state-of-the-art [2,11,12,3,13] on standard scalable multimodal benchmark problems, that are characterized by different difficulties in local optimality, non-uniformity, discontinuity, non-convexity and high-dimensionality. The comprehensive experimental investigations shows the effectiveness of the proposed algorithm in terms of convergence, quality of solution and robustness. The competitiveness of the proposed algorithm is also demonstrated on dynamic optimization problems.

The paper is organized as follows: Section 2 illustrate basic artificial bee colony (ABC) algorithm. Section 3 describes DABC algorithm and external archive. Section 4 presents experimental setup adopted for performance comparison. The results are discussed in Section 5 and Section 6 concludes the paper.

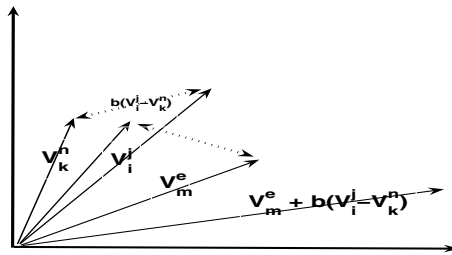


Fig. 1. Figure illustrating the vector addition of differential update strategy

2 Artificial Bee Colony (ABC) Algorithm

In 2005 Karaboga has introduced an artificial be colony (ABC) algorithm [4]. The performance of ABC is analyzed in 2008 [4]. The ABC algorithm was developed to mimic the behaviors of the real bees on finding food source (the nectar) and sharing the information of the nectar to the other bees in the hive. The colony of bees are classified as employed, onlooker, and the scout bees. Each of them plays different role in the search process: The first half of the colony

consists of the employed bees and the second half includes the onlookers. For every food source, there is only one employed bee. The number of employed bees is equal to the number of food sources around the hive. The employed bee stays on a food source and remember in its memory; the onlooker bees get the information of food sources from the employed bees in the hive and select one of the food source to gather the nectar. The employed bee whose food source has been exhausted becomes a scout. The scout is responsible for finding new food sources independently. The information exchange among bees is through a special dance called waggle dance. Since information about all the current rich sources is available to onlookers on the dance floor and hence they choose the food source with higher fitness. Employed foragers share their information with a probability, which is proportional to the fitness of the food source, and hence, the recruitment is proportional to profitability of a food source [4,9]. The bees search for the food sources in order to maximize of the objective function $F(\theta)$, $\theta \in R$ (for maximization problem). The θ_i is the position of the i^{th} food source and $F(\theta_i)$ the fitness of the nectar amount located at θ_i . The waggle dance of the employed bees decides about where the onlooker bees go for foraging. The onlooker bee goes to the region of food source located at θ_i by probability [9] and determines food sources around. The i^{th} dimension of j^{th} bee is updated as

$$\theta_i^j(c+1) = \theta_i^j(c) + rand * (\theta_i^j(c) - \theta_k^n(c)) \quad (1)$$

Where c corresponds to the current iteration, $\theta_k^n(c)$ is the randomly selected k^{th} dimension of randomly selected n^{th} neighbor. $rand$ is the random number in $[0,1]$. The $\theta_i(c)$ is changed to $\theta_i(c+1)$ if the nectar amount $F(\theta_i(c+1))$ at $\theta_i(c+1)$ is higher than that at $\theta_i(c)$, else $\theta_i(c)$ is unchanged. If the position θ_i of the food source ' i ' cannot be improved for a predetermined number of attempts; say " $limit$ ", then it is abandoned by its employed bee and becomes a scout. The scout starts to search for a new food source randomly.

3 Differential Artificial Bee Colony (DABC)

The major enhancement that can be made to ABC algorithm is the bees update strategy. In classical ABC algorithm onlooker bees and employed bees update their coordinates based on the information of their neighborhood bees (equation 1). The best solution that is being produced in every iteration during search process is only memorized and used as retention candidate. The ABC algorithm does not utilize the capability of potential solution being produced for better searching. The classical ABC algorithm even lacks the ability to retain a good number of potential solutions produced through the generations. This paper proposes a novel strategy based on differential update and the external achieve. The major enhancements that are being incorporated are discussed under the following headings.

3.1 Differential Update Strategy with Elite Bee

In the differential update strategy the elite bee is selected in every iteration. The elite bee has the best possible solution to the problem under consideration in

the history of search process. According to the differential update strategy, the weighted difference of the bee and its neighbor are added under the guidance of the elite bee as shown in Fig 1. The i^{th} dimension of j^{th} bee is updated as

$$\theta_i^j(c+1) = \theta_m^e(c) + \beta * (\theta_i^j(c) - \theta_k^n(c)) \quad (2)$$

Where c corresponds to the current iteration, $\theta_m^e(c)$ is the randomly selected m^{th} dimension of the elite bee, and, $\theta_k^n(c)$ the randomly selected k^{th} dimension of randomly selected n^{th} neighbor. The β is the weight, in our experiment we have used $\beta = 0.5$.

3.2 Preserving Good Solutions: The Archive

To preserve the good solutions produced in search, an external archive is maintained. Initially, the archive is empty. The size of the archive is same as the number of colony size in this paper. As the generation progresses, good solutions enter the archive and is updated in every generation. The good solutions obtained in each generation are compared one by one with the current archive and if new solution is better than the members of the archive, then it enters the archive. If archive exceeds predefined size then poor quality solutions rejected.

3.3 Differential Update Strategy with Exemplars of Archive

Since an external archive is maintained to retain the number of good solutions produced, theoretically the exemplars from the archive are the best in history. Again here, according to the differential update strategy, the weighted difference of the bee and its neighbor are added under the guidance of randomly selected exemplars of archive as shown in Fig 1. The i^{th} dimension of j^{th} bee is updated as follows

$$\theta_i^j(c+1) = \theta_m^a(c) + \beta * (\theta_i^j(c) - \theta_k^n(c)) \quad (3)$$

Where $\theta_m^a(c)$ is the randomly selected m^{th} dimension of the randomly selected a^{th} exemplar of the archive. The meaning of c , $\theta_k^n(c)$ and β are same as above in equation 2.

4 Simulation

4.1 Parameter Settings and PC Configuration

The experiments were carried out on Pentium IV, 2GHz with 1GB RAM. The programs were written in Matlab 7.2 on Windows-XP platform. All the algorithms are set with the population size of 25, the number of iterations 1000. The results recorded are the average of 20 trials. The stopping criteria for all the algorithms is set to 1×10^{-250} . The dimensions of 10, 30 and 50 were used for testing quality of results. For *DABC* external archive size is set equal to the population, the weight $\beta = 0.5$ and the other parameters are same as that of classical ABC [8].

Algorithmic parameters for PSO and HAPSO are as in [12].

Algorithm 1. Differential Artificial Bee Colony (DABC))

Initialize the colony and Archive
 Evaluate the nectar amounts of employed bees.
 Find the best bee among the bee swarm: elite.
 Calculate the probability of selecting a food source as in [9]
 Select a food source to move by roulette wheel for every onlooker bees.
 Move the onlookers under the guidance of elite equation 2.
 Find the exhausted employed bee: the scout.
 Move the scouts randomly in the search range.
 Move the bees under the guidance of exemplars from archive equation 3.
 Select the good bees from the hive
 Update the existing archive as in section 3.2
 if Archive exceeds maximum size eliminate poor nectar bees.
 continue optimizing until stopping criteria or exceeding maximum iteration

Table 1. Benchmark Functions Definition

Function	Function definition	Asymmetric search range
$f_1(\vec{x})$	$20 + e^{-\frac{1}{5}\sqrt{\frac{1}{D}\sum_i(x_i)^2}} - e^{-\frac{1}{D}} \sum_i \cos(2\pi x_i)$	$-16.384 \leq x_i \leq 32.768$
$f_2(\vec{x})$	$\sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$-300 \leq x_i \leq 600$
$f_3(\vec{x})$	$\sum_{i=1}^D (x_i^2 - 10\cos(2\pi x_i) + 10)$	$-2.560 \leq x_i \leq 5.12$
$f_4(\vec{x})$	$\sum_{i=1}^{D-1} \left[100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2 \right]$	$-15 \leq x_i \leq 30$
$f_5(\vec{x})$	$\sum_{i=1}^D x_i^2$	$-50 \leq x_i \leq 100$
$f_6(\vec{x})$	$\sum_{i=1}^D x_i^2 + \left(\sum_{i=1}^D 0.5ix_i\right)^2 + \left(\sum_{i=1}^D 0.5ix_i\right)^4$	$-5 \leq x_i \leq 10$
$f_7(\vec{x})$	$(x_1 - 1)^2 + \sum_{i=2}^D i(2x_i^2 - x_i - 1)^2$	$-5 \leq x_i \leq 10$

f_8 to f_{14} are shifted versions of f_1 to f_7 respectively

4.2 Algorithms

Particle Swarm Optimization (PSO): PSO was introduced in 1995 by Kennedy and Eberhart [2]. PSO simulates the social life, such as a swarm of birds or school of fish [11]. The velocity and position of each particle is updated by the following equations respectively [11].

A Hierarchical Particle Swarm Optimizer and Its Adaptive Variant: A Hierarchical Particle Swarm Optimizer and Its Adaptive Variant is a novel method in PSO called HAPSO here. The particles in HAPSO [12] are arranged in a dynamic hierarchy that is used to define a neighborhood structure. Particles move up and down the hierarchy with the effect of best-found solution.

4.3 Benchmark Functions

The comprehensive performance analysis of the algorithms are carried out on the two sets of well-known standard benchmark functions. All the selected functions are complex, multimodal and scalable in dimensions. The first set consists of

normal functions Ackley, Griewank, Rastrigin, Rosenbrock, Sphere, Zakharov and Dixon Price, they are numbered as f_1 to f_7 respectively in Table 1. The second set consists of pseudo dynamic functions, i.e. they are the shifted versions of the first set and they are numbered in our experiment as f_8 to f_{14} respectively. The functions are shifted by adding the random bias to the original value \vec{x} to obtain the new shifted variable $\vec{y} = \vec{x} - O$. This variable \vec{y} is used to calculate the fitness value. The random bias is generated in the search range for every run (trail) and the algorithms are allowed to search for that run as follows.

$$O = R_{min} + rand * (R_{max} - R_{min})$$

Where R_{min} and R_{max} are lower and upper bounds and $rand$ is uniform random number in the range [0 1].

5 Results

The experiments were conducted on a set of 14 (normal and shifted) complex multimodal scalable benchmark problems of dimensions 10, 30 and 50 with asymmetric initialization. The performance comparisons are done with three different aspects a) mean results and b) robustness.

5.1 Mean Results

The average results are presented from Table 2 to Table 4. The first two columns of these tables shows the functions and the algorithms. Table 2 to Table 4 gives the complete information about the mean result and standard deviations achieved in sixth and seventh columns. The best mean result and best standard deviations achieved by the algorithms are shown in bold. The mean results of 10 dimension problems are shown in Table 2. The DABC performs well on almost all the problems f_1 to f_{14} with 10D problems. The worst results obtained by DABC are better than the best results obtained by the other algorithms on almost all problems except on f_4 and f_7 where HAPSO does well. Table 3 records the 30D results, it can be observed from this table that the mean results of DABC performs well on almost all the problems. From the same table worst results obtained by DABC are better than the best results obtained by the other algorithms on almost all problems except on f_4 and f_6 where HAPSO does well. The Table 4 shows the better performance of DABC on 50D problems except f_6 where HAPSO obtain good result. From Table 4 once again it is observed that the worst results obtained by DABC are better than the best results obtained by the other algorithms on almost all problems except on f_4 , f_6 and f_7 where HAPSO does well again.

5.2 Robustness

Since almost all the swarm intelligence algorithms are stochastic in nature, they do not produce the same results every time they are run, and hence a stability test is required. The stability or robustness of the algorithms are represented

Table 2. Optimum values achieved with functions of dimension=10

func	alg	best	worst	median	mean	std
f_1	PSO	1.30e+1	1.93e+1	1.60e+1	1.59e+1	2.02
	ABC	4.25e-1	1.91	1.18	1.14	5.05e-1
	HAPSO	1.21e-5	3.45e-4	9.87e-5	1.12e-4	9.21e-5
	DABC	8.88e-16	4.44e-15	4.44e-15	3.73e-15	1.50e-15
f_2	PSO	2.54e+1	7.63e+1	3.79e+1	4.44e+1	1.71e+1
	ABC	1.59e-1	4.74e-1	2.90e-1	3.00e-1	9.01e-2
	HAPSO	3.49e-10	6.64e-9	1.23e-9	1.80e-9	1.93e-9
	DABC	0.0	0.0	0.0	0.0	0.0
f_3	PSO	6.04e+1	1.10e+2	8.55e+1	8.78e+1	1.89e+1
	ABC	9.10e-1	5.46	3.20	3.34	1.16
	HAPSO	6.34e-8	7.04e-6	2.46e-6	2.88e-6	2.16e-6
	DABC	0.0	0.0	0.0	0.0	0.0
f_4	PSO	5.11e+5	9.23e+6	3.03e+6	4.02e+6	3.15e+6
	ABC	3.23e+1	9.43e+1	5.56e+1	5.67e+1	1.80e+1
	HAPSO	5.32	1.60e+1	8.68	9.13	2.66
	DABC	1.93e-16	5.99	4.21e-8	1.74	2.38
f_5	PSO	3.01e+3	7.04e+3	4.56e+3	4.78e+3	1.20e+3
	ABC	1.01e-3	2.89e-2	1.66e-2	1.55e-2	9.74e-3
	HAPSO	1.71e-9	1.03e-7	6.69e-9	1.83e-8	3.09e-8
	DABC	4.91e-21	2.31e-19	6.13e-20	7.17e-20	6.78e-20
f_6	PSO	1.04e+2	2.23e+2	1.44e+2	1.56e+2	4.15e+1
	ABC	5.47	3.20e+1	2.22e+1	2.16e+1	7.45
	HAPSO	6.57e-8	2.69e-6	6.21e-7	8.07e-7	7.86e-7
	DABC	4.79e-20	4.33e-18	9.82e-19	1.44e-18	1.18e-18
f_7	PSO	1.42e+3	1.97e+4	8.31e+3	9.54e+3	6.11e+3
	ABC	2.90e-1	1.03	5.88e-1	6.52e-1	2.62e-1
	HAPSO	4.32e-1	9.39e-1	6.79e-1	6.64e-1	1.44e-1
	DABC	3.79e-7	6.67e-1	3.33e-1	3.33e-1	3.51e-1
f_8	PSO	1.27e+1	1.99e+1	1.71e+1	1.69e+1	2.07
	ABC	5.85e-1	3.24	1.13	1.60	9.24e-1
	HAPSO	2.49	1.56e+1	8.76	8.72	4.21
	DABC	8.88e-16	5.77e-14	8.88e-16	7.64e-15	1.77e-14
f_9	PSO	1.64e+1	2.19e+2	5.32e+1	6.73e+1	6.00e+1
	ABC	1.27e-1	5.14e-1	3.34e-1	3.42e-1	1.26e-1
	HAPSO	1.77e-2	8.00e+1	2.74	1.42e+1	2.49e+1
	DABC	0.0	0.0	0.0	0.0	0.0
f_{10}	PSO	5.88e+1	1.49e+2	1.08e+2	1.08e+2	2.98e+1
	ABC	6.69e-1	8.19	5.57	5.31	2.19
	HAPSO	1.07e+1	4.80e+1	2.89e+1	2.81e+1	9.62
	DABC	0.0	0.0	0.0	0.0	0.0
f_{11}	PSO	1.03e+6	7.26e+7	1.44e+7	2.25e+7	2.49e+7
	ABC	1.27e+1	1.23e+2	3.35e+1	4.18e+1	3.12e+1
	HAPSO	9.61	1.27e+5	5.90e+3	2.65e+4	4.30e+4
	DABC	3.70e-18	8.15	4.44	3.38	3.11
f_{12}	PSO	4.24e+3	1.88e+4	8.90e+3	9.72e+3	3.95e+3
	ABC	2.36e-3	8.45e-2	1.10e-2	1.97e-2	2.49e-2
	HAPSO	4.40	6.50e+3	3.98e+2	1.22e+3	2.07e+3
	DABC	8.70e-21	1.31e-19	3.62e-20	5.48e-20	4.72e-20
f_{13}	PSO	5.08e+1	3.27e+7	1.29e+4	5.74e+6	1.04e+7
	ABC	1.87	3.10e+1	1.67e+1	1.77e+1	9.62
	HAPSO	5.51e-1	6.07e+4	6.60	6.08e+3	1.92e+4
	DABC	1.76e-19	2.30e-18	1.76e-18	1.53e-18	7.78e-19
f_{14}	PSO	2.07e+3	1.50e+5	5.24e+3	3.94e+4	5.95e+4
	ABC	3.18e-1	2.08	6.10e-1	8.69e-1	5.93e-1
	HAPSO	1.83e-1	1.33e+2	1.35e+1	4.24e+1	5.10e+1
	DABC	1.93e-7	6.67e-1	6.29e-5	2.00e-1	3.22e-1

Table 3. Optimum values achieved with functions of dimension=30

func	alg	best	worst	median	mean	std
f_1	PSO	1.90e+1	1.99e+1	1.95e+1	1.95e+1	3.06e-1
	ABC	5.42	7.37	5.85	6.09	7.42e-1
	HAPSO	6.64e-5	2.52e-4	1.33e-4	1.39e-4	6.09e-5
	DABC	7.99e-15	3.29e-14	2.22e-14	2.29e-14	6.44e-15
f_2	PSO	2.74e+2	4.64e+2	3.76e+2	3.66e+2	5.77e+1
	ABC	1.11	2.15	1.46	1.51	3.20e-1
	HAPSO	1.15e-9	1.20e-8	3.09e-9	4.12e-9	3.32e-9
	DABC	0.0	0.0	0.0	0.0	0.0
f_3	PSO	3.44e+2	3.95e+2	3.79e+2	3.75e+2	1.52e+1
	ABC	4.37e+1	6.88e+1	6.21e+1	6.02e+1	7.42
	HAPSO	1.62e-6	2.10e-5	5.38e-6	7.90e-6	6.47e-6
	DABC	0.0	0.0	0.0	0.0	0.0
f_4	PSO	3.26e+7	1.44e+8	1.24e+8	1.06e+8	4.05e+7
	ABC	1.30e+3	1.00e+4	3.99e+3	5.12e+3	2.63e+3
	HAPSO	2.14e+1	2.87e+1	2.87e+1	2.79e+1	2.31
	DABC	1.92e-12	2.77e+1	1.37e+1	1.38e+1	1.45e+1
f_5	PSO	3.72e+4	4.79e+4	4.32e+4	4.22e+4	4.29e+3
	ABC	1.06e+1	6.95e+1	2.42e+1	2.83e+1	1.78e+1
	HAPSO	7.54e-9	2.26e-7	5.24e-8	6.61e-8	6.57e-8
	DABC	2.46e-18	1.47e-17	9.73e-18	9.26e-18	4.12e-18
f_6	PSO	4.80e+2	3.45e+9	1.34e+7	6.72e+8	1.19e+9
	ABC	2.29e+2	3.12e+2	2.67e+2	2.67e+2	2.84e+1
	HAPSO	1.25e-6	5.13e-5	7.66e-6	1.75e-5	1.80e-5
	DABC	1.22e-6	2.37e-5	3.80e-6	7.10e-6	7.76e-6
f_7	PSO	2.14e+5	9.85e+5	5.29e+5	5.92e+5	2.82e+5
	ABC	2.18e+1	1.66e+2	6.66e+1	8.46e+1	5.64e+1
	HAPSO	8.99e-1	9.42e-1	9.32e-1	9.26e-1	1.51e-2
	DABC	6.67e-1	6.67e-1	6.67e-1	6.67e-1	3.60e-13
f_8	PSO	1.85e+1	2.13e+1	1.99e+1	2.01e+1	9.07e-1
	ABC	4.62	1.32e+1	7.67	8.33	3.27
	HAPSO	5.49	1.56e+1	1.37e+1	1.29e+1	2.85
	DABC	7.99e-15	2.93e-14	1.51e-14	1.58e-14	6.86e-15
f_9	PSO	2.27e+2	1.17e+3	3.73e+2	4.99e+2	2.97e+2
	ABC	1.07	1.65	1.22	1.28	1.94e-1
	HAPSO	1.12	3.87e+2	2.37e+1	6.81e+1	1.15e+2
	DABC	0.0	0.0	0.0	0.0	0.0
f_{10}	PSO	3.01e+2	6.53e+2	3.66e+2	4.52e+2	1.41e+2
	ABC	5.08e+1	1.00e+2	6.70e+1	6.91e+1	1.48e+1
	HAPSO	4.60e+1	3.08e+2	1.41e+2	1.44e+2	7.57e+1
	DABC	0.0	0.0	0.0	0.0	0.0
f_{11}	PSO	7.88e+7	4.08e+8	1.39e+8	1.72e+8	1.08e+8
	ABC	9.25e+2	8.19e+3	1.75e+3	3.38e+3	2.95e+3
	HAPSO	8.27e+2	2.95e+6	5.39e+4	7.20e+5	1.18e+6
	DABC	9.86e-10	2.84e+1	2.74e+1	1.93e+1	1.33e+1
f_{12}	PSO	3.55e+4	1.15e+5	5.44e+4	6.45e+4	3.21e+4
	ABC	8.14	1.29e+2	2.48e+1	3.84e+1	3.71e+1
	HAPSO	8.39	1.32e+4	5.70e+3	5.86e+3	4.00e+3
	DABC	3.88e-18	1.40e-17	5.16e-18	6.59e-18	3.25e-18
f_{13}	PSO	7.84e+3	2.5012	1.66e+9	2.8211	7.8511
	ABC	9.31e+1	2.89e+2	2.27e+2	2.12e+2	6.67e+1
	HAPSO	3.14e-1	1.73e+6	3.55e+1	1.73e+5	5.46e+5
	DABC	4.48e-8	7.02e-5	3.95e-6	1.71e-5	2.53e-5
f_{14}	PSO	1.90e+5	7.67e+6	9.67e+5	2.18e+6	2.37e+6
	ABC	7.83	4.80e+2	3.74e+1	8.83e+1	1.44e+2
	HAPSO	1.01e+1	1.58e+5	1.64e+3	2.06e+4	4.89e+4
	DABC	1.51e-3	6.72e-1	6.67e-1	5.35e-1	2.79e-1

Table 4. Optimum values achieved with functions of dimension=50

func	alg	best	worst	median	mean	std
f_1	PSO	1.96e+1	2.05e+1	2.02e+1	2.01e+1	2.90e-1
	ABC	7.59	1.20e+1	1.04e+1	1.01e+1	1.47
	HAPSO	4.51e-5	2.30e-4	1.43e-4	1.33e-4	5.54e-5
	DABC	9.68e-14	3.35e-13	1.79e-13	1.86e-13	7.64e-14
f_2	PSO	5.69e+2	8.43e+2	7.45e+2	7.41e+2	9.40e+1
	ABC	2.56	1.15e+1	4.44	5.41	3.03
	HAPSO	8.19e-10	7.53e-9	3.13e-9	3.24e-9	2.02e-9
	DABC	00	1.33e-15	3.33e-16	4.22e-16	4.31e-16
f_3	PSO	5.99e+2	7.00e+2	6.63e+2	6.62e+2	2.96e+1
	ABC	1.40e+2	1.87e+2	1.64e+2	1.65e+2	1.83e+1
	HAPSO	1.26e-6	3.73e-5	7.81e-6	1.03e-5	1.02e-5
	DABC	0.0	0.0	0.0	0.0	0.0
f_4	PSO	1.41e+8	2.98e+8	2.43e+8	2.30e+8	4.92e+7
	ABC	2.10e+4	2.14e+5	5.74e+4	8.11e+4	6.64e+4
	HAPSO	3.56e+1	5.19e+1	4.85e+1	4.76e+1	4.33
	DABC	4.48e-8	4.77e+1	4.76e+1	3.33e+1	2.30e+1
f_5	PSO	7.17e+4	8.85e+4	8.20e+4	8.04e+4	5.86e+3
	ABC	9.91e+1	8.21e+2	4.06e+2	4.40e+2	2.29e+2
	HAPSO	8.75e-9	1.77e-7	4.02e-8	5.48e-8	5.20e-8
	DABC	1.57e-17	1.76e-16	5.96e-17	6.78e-17	4.34e-17
f_6	PSO	1.4110	4.5211	1.1611	1.2811	1.3011
	ABC	5.53e+2	6.36e+2	5.85e+2	5.91e+2	3.20e+1
	HAPSO	3.62e-6	2.17e-3	1.15e-4	3.78e-4	6.58e-4
	DABC	7.98e-3	1.92e-1	3.25e-2	5.39e-2	5.71e-2
f_7	PSO	1.61e+6	4.03e+6	2.75e+6	2.77e+6	7.92e+5
	ABC	2.51e+2	1.78e+3	9.96e+2	9.98e+2	4.81e+2
	HAPSO	9.72e-1	9.89e-1	9.84e-1	9.83e-1	4.88e-3
	DABC	1.76e-1	6.67e-1	6.67e-1	5.69e-1	2.06e-1
f_8	PSO	1.94e+1	2.13e+1	2.08e+1	2.05e+1	7.98e-1
	ABC	9.11	1.98e+1	1.56e+1	1.49e+1	4.87
	HAPSO	1.02e+1	1.89e+1	1.42e+1	1.49e+1	3.03
	DABC	8.62e-14	1.02e-8	1.41e-13	1.02e-9	3.23e-9
f_9	PSO	5.61e+2	2.19e+3	9.66e+2	1.28e+3	6.35e+2
	ABC	3.14	4.17e+1	5.46	1.12e+1	1.18e+1
	HAPSO	6.45	6.93e+2	2.34e+2	3.00e+2	2.52e+2
	DABC	00	3.33e-16	2.22e-16	1.89e-16	1.48e-16
f_{10}	PSO	6.59e+2	1.15e+3	8.39e+2	8.50e+2	1.56e+2
	ABC	1.32e+2	2.79e+2	1.91e+2	1.97e+2	4.79e+1
	HAPSO	3.10	3.86e+2	2.92e+2	2.68e+2	1.11e+2
	DABC	00	1.71e-13	5.68e-14	4.55e-14	5.22e-14
f_{11}	PSO	3.43e+8	2.51e+9	8.81e+8	1.03e+9	6.78e+8
	ABC	1.27e+4	4.06e+6	6.58e+4	5.53e+5	1.25e+6
	HAPSO	1.23e+3	2.16e+6	4.26e+5	6.63e+5	7.68e+5
	DABC	1.26e-11	1.01e+2	4.76e+1	3.87e+1	3.14e+1
f_{12}	PSO	7.39e+4	1.82e+5	1.13e+5	1.18e+5	4.08e+4
	ABC	8.82e+1	1.69e+3	5.60e+2	7.19e+2	4.72e+2
	HAPSO	7.21e+2	4.70e+4	1.23e+4	1.61e+4	1.40e+4
	DABC	3.94e-17	1.57e-16	5.90e-17	7.09e-17	3.51e-17
f_{13}	PSO	2.64e+8	5.3213	2.2410	1.1713	2.1613
	ABC	1.14e+2	5.27e+2	2.67e+2	2.84e+2	1.32e+2
	HAPSO	2.71	1.8112	7.56e+2	2.0511	5.6811
	DABC	2.59e-2	1.32e+1	5.09e-2	1.37	4.16
f_{14}	PSO	3.96e+6	2.61e+7	8.62e+6	1.07e+7	7.21e+6
	ABC	1.94e+2	2.38e+3	5.41e+2	7.49e+2	6.69e+2
	HAPSO	2.95e+2	2.19e+5	9.73e+3	3.21e+4	6.69e+4
	DABC	1.66e-2	6.68e-1	6.67e-1	6.02e-1	2.06e-1

by the standard deviation they achieve over 20 trials and 1000 iterations. A **0.0** value of standard deviation indicate very stable nature of the algorithm, any other value indicate the deviation. The best standard deviation is shown with bold. From Table 2 it can be observed that DABC is stable on all of the problems for dimension 10, except f_4 where HAPSO is stable. The Table 3 shows the better stability of DABC on almost all problems for dimension 30, except f_4 where HAPSO is stable. The Table 4 that records the results of 50D problems shows the stable nature of DABC on all problems except the problems f_4 , f_6 and f_7 where again HAPSO shows its stability.

5.3 Mean Result Comparison with Other State-of-the-Art

The mean result of the HAPSO [12], CSA [3], CSADE [3] and HPSO [12] are compared with DABC in Table 5. From Table 5 it can be concluded that DABC produces the competitive results on functions f_1 to f_5 of 50 dimension. The HPSO alone shows the better results on function f_5 .

Table 5. Solutions achieved by DABC and other state-of-the-art on 50D problems

func	CSA [3]	CSADE [3]	HAPSO [12]	HPSO [13]	DABC
f_1	19.7493	1.0419e-4	1.39e-4	2.53	2.29e-14
f_2	135.7764	115.0552	4.12e-9	-	0.0
f_3	455.9070	9.4486e-6	7.90e-6	33.5	0.0
f_4	1.5870e+6	48.8484	2.79e+1	19.7	1.38e+1
f_5	7.6837e+4	4.5101e-4	6.61e-8	4.20e-43	9.26e-18

6 Conclusion

This paper addresses the problems associated with the bee update strategy in classical artificial bee colony (ABC) algorithm. The novel differential update strategy and external archive are introduced with the classical ABC algorithm to improve the convergence speed and quality of the solution. Further this paper presents a comprehensive investigation and analysis of developed variant against well known stochastic algorithms on a set of standard multimodal benchmark functions and their shifted versions. Results show that the performance of the proposed algorithm is better and competitive. The notable feature of the proposed algorithm lies in its stability in finding the solution even for the dynamic optimization problems. The mean result achieved by the proposed algorithms does not effect much as the dimensions of the problem increased. The authors are working towards application of this algorithm to the varied engineering design problems especially in sensor networks.

References

1. Alok, S.: An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem. *Applied Soft Computing* 9(2), 625–631 (2009)
2. Eberhart, R., Kenedy, J.: Particle swarm optimization. In: *Proceedings of IEEE Int. Conference on Neural Networks*, Piscataway, NJ, pp. 1114–1121 (November 1995)
3. Gao, X.Z., Wang, X., Ovaska, S.J.: Fusion of clonal selection algorithm and differential evolution method in training cascadecorrelation neuralnetwork. *Neurocomputing* 72(2), 2483–2490 (2009)
4. Karaboga, D.: An idea based on honey bee swarm for numerical optimization. Tech. Rep. TR06, Computer Engineering Department, Engineering Faculty, Erciyes University, Turkey (2005)
5. Karaboga, D.: A new design method based on artificial bee colony algorithm for digital iir filters. *Journal of The Franklin Institute* 346(4), 328–348 (2009)
6. Karaboga, D., Basturk, B.: An artificial bee colony (abc) algorithm for numeric function optimization. In: *IEEE Swarm Intelligence Symposium*
7. Karaboga, D., Basturk, B.: An artificial bee colony (abc) algorithm on training artificial neural networks. In: *15th IEEE Signal Processing and Communications Applications*, Eskisehir, Turkiye, pp. 1–4 (June 2007)
8. Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (abc) algorithm. *Journal of Global Optimization* 39(3), 459–471 (2007)
9. Karaboga, D., Basturk, B.: On the performance of artificial bee colony (abc) algorithm. *Applied Soft Computing* 8(1), 687–697 (2008)
10. Karaboga, D., Basturk, B., Ozturk, C.: Artificial bee colony (abc) optimization algorithm for training feed-forward neural networks. In: Torra, V., Narukawa, Y., Yoshida, Y. (eds.) *MDAI 2007. LNCS (LNAI)*, vol. 4617, pp. 318–329. Springer, Heidelberg (2007)
11. Robinson, J., Rahmath Samii, Y.: Particle swarm optimization in electromagnetic. *IEEE Transactions on Antenna and Propagation* 52(2), 397–400 (2004)
12. Stefan, J., Martin, M.: A hierarchical particle swarm optimizer and its adaptive variant. *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics* 35(6), 1272–1282 (2005)
13. Zhigang, Z.: Modified particle swarm optimization for unconstrained optimization. In: *The 2nd International Conference on Computer and Automation Engineering (ICCAE)*, Chongqing, China, pp. 377–380 (February 2010)

FAM2BP: Transformation Framework of UML Behavioral Elements into BPMN Design Element

Jayeeta Chanda¹, Ananya Kanjilal¹, Sabnam Sengupta¹, and Swapan Bhattacharya²

¹ B.P. Poddar Institute of Management & Technology, Kolkata -52
jayeeta.chanda@gmail.com
ag_k@rediffmail.com
sabnam_sg@yahoo.com

² National Institute of Technology, Durgapur, 713209, West Bengal
bswapan2000@yahoo.co.in

Abstract. Business processes are an integral part of service-oriented architecture. A typical business process spans multiple Use Cases. Use case diagram along with activity diagrams represents the behavior of a design in the analysis phase for an object-oriented system. In this paper, we propose a relational model, "Formalized Analysis Model to Business Process (FAM2BP)" for transformation of Formalized Analysis Model (FAM) of object-oriented systems into BPMN process for SOA application. FAM consists of a set of grammar based formalized Use case and Activity diagram elements of UML. FAM2BP propose some rules that will help the designer to generate business processes for SOA application directly from object oriented analysis models. This model would help in evolution of software design and development paradigms from Object Oriented to Service Oriented systems.

Keywords: Service-oriented-architecture, BPMN-Process, Relational-model, Translation from OO to SOA, UML to BPMN, Formal UML model.

1 Introduction

Design and development of software has become much more complex in the last decade, resulting in evolution of design and development paradigms. Object oriented systems have thus become an integral part of more complex Service Oriented Architecture (SOA) to address complex issues like Separation of Concerns, reusability, granularity, modularity, componentization and interoperability. Evolution of software design and development from object oriented to SOA domain has become the necessity in this evolving scenario. We already have some proven Object Oriented design tools that can be used for SOA application design. In software projects, Use case diagrams and activity diagrams are used to model the business functional requirements and its flow of events in the analysis phase. These correlate to the business processes of SOA architecture. In service-oriented architecture, BPMN processes play an important role in the development of services. Automatic translation of UML use case and activity models to BPMN design elements would ensure consistent evolution of Object oriented systems to Service oriented paradigm. In this paper, we propose a

relational model based framework to address this need. The automatic transformation helps the designer to develop the service oriented architecture in a more convenient manner as well as ensuring consistency and correctness in the design.

2 Review of Related Work

There exists some work related to the relationship between use case models and BPMN process model. In [1], Cockburn mentions the possibility of applying Use Cases for deriving business processes but no rules are proposed. The field of model-driven development has tried to integrate the concept of Use Cases within its UML models. Instead of tabular and textual descriptions, UML sequence diagrams or similar models are used [4]. Use case diagrams are used to define use cases and other models are used to define scenario. In [6], an UML based development of business process is discussed. Expression of control flow between use cases is missing in this approach.

In [9], it is possible to define control-flow dependencies between Use Cases with the introduction of Use Case Charts and their formalization. Use Cases are called scenarios that may not have extensions and that are modeled as UML sequence diagrams. However, dependencies between Use Cases cannot be derived from the Use Case themselves but have to be modeled explicitly.

In [7], synthesis of state transition graphs from Use Cases is better addressing the visualization aspect. A tabular Use Case can be converted to a state transition graph similar to graphical business process languages. In [8], the state transition graphs can be used for simulating one Use Case but are not suited for visualizing dependencies between Use Cases. The generation of EPC models from Use Cases is addressed in [3]. EPC models consist of fewer graphical symbol types but are not as powerful as BPMN. BPMN has become the standard business process modeling language in SOA. Therefore, the transformation of Use Cases should have BPMN as the target notation and our framework is developed upon this concept.

In [10], an algorithm is proposed that restores the overview of the Use cases and visualizes the control flow of the resulting business process. This approach automatically assembles Use cases to business process. But this work is unable to keep the relationship (includes and extends) among use cases and treated as flat use case model.

Our work is closely related to these works but improves upon them in several aspects. We capture the use case scenarios as a formalized analysis model (FAM) that is a grammar based representation of UML use case and activity models. A formal definition of semantics for the subset of BPMN that is applied in this paper has been presented in [2]. Then a set of rules are proposed that automatically transforms the FAM to BPMN elements maintaining the control flow of scenarios as well as preserving all relationships between the use cases.

3 Scope of Work

In this paper we propose a framework for automatic transformation of UML analysis models to BPMN design elements for Service Oriented paradigm.. In this framework the elements of formalized analysis model (FAM) are transformed into the elements of business processes. FAM consists of some grammar based formalized Use case and Activity diagram elements of UML as done in our work [11]. The Framework consists

of a set of rules to map the UML elements like events and flow of events into BPMN elements like - start/stop/intermediate events, parallel/exclusive-OR Gateway, etc. A relational model is proposed to represent the relationship between all the artifacts. Finally an algorithm is presented to automatically transform the UML elements into BPMN elements. The block diagram in figure1 depicts our approach.

4 FAM2BP: Proposed Transformation Model

The input to our model is the formal analysis model (FAM) which is presented in the following section 4.1. The next section 4.2 discusses the relational model which maps the artifacts of the two paradigms. The transformation rules are presented in 4.3 and the algorithms for automatic transformation are presented in 4.4.

4.1 Formalized Analysis Model

As proposed in our work [11], we are using the Formalized Analysis Model that consists of context free grammar for the Use Case and Activity Diagrams. The grammars are as proposed in our work [11].

4.2 The Relational Model

As shown in figure1, the elements of Formalized Analysis Model (FAM) are mapped with the BPMN node. For example, the *Usecase* and *Event* entity are related to each other by *have* relation. Similarly all other elements of the FAM are related as shown in Figure 2. These elements of FAM model are mapped with the BPMN node to ensure automatic transformation from Formalized Analysis Model to Business process. The tables corresponding to this model are generated in Figure 2.

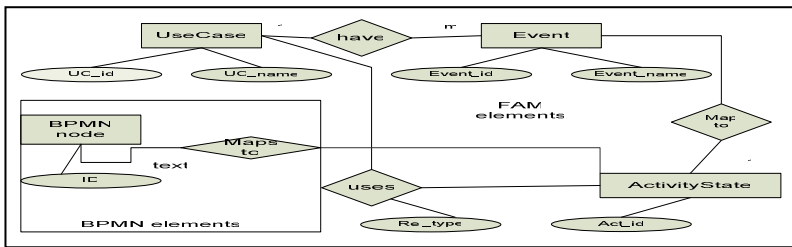


Fig. 1. The proposed Relational Model between different elements of FAM

4.3 Transformation Rules

We propose a set of rules to transform Formalized Analysis model into BPMN Notation.

The rules are the defined as follow:

Rule1

The use case activity whose node is marked as ‘start’ will be assigned as Start Event of the BPMN node. The BPMN node will labeled as Activity ID (act_ID) of the activity node.






ID	Name	Activity node	Graphical notation	label
1	Start event	Start		
2	End Event	End		
3	Intermediate Event	Action, decision		
4	Parallel Gateway	Fork, join		
5	Exclusive-OR Gateway	Fork, join		

Fig. 2. Relational Model for FAM in tabular form

Rule 2

The use case activity whose node is marked as ‘end’ will be assigned as End Event of the BPMN node. The BPMN node will be labeled as Activity ID (act_ID) of the activity node.

Rule3

The use case activity whose node is marked as ‘action / decision’ will be assigned as Intermediate Event of the BPMN node. The BPMN node will be labeled as Activity ID (act_ID) of the activity node.

Rule 4

The use case activity whose node is marked as ‘fork’ will be assigned as Parallel Gateway of the BPMN node if both the postElement of the activity node are of the type ‘basic’. The BPMN node will be labeled as Activity ID (act_ID) of the activity node.

Rule 5

The use case activity whose node is marked as ‘fork’ will be assigned as Exclusive-OR Gateway of the BPMN node if one the postElement of the activity node are of the type ‘basic’ and the other is of the type ‘alternate’. The BPMN node will be labeled as Activity ID(act_ID) of the activity node.

These rules are realized in the next section to automate the transformation of Formalized Analysis Model into BPMN nodes.

4.4 Algorithm for Automated Transformation

These rules cited in section 4.3 are realized using two algorithms namely *NodeGeneration* and *FlowGeneration*. The flow of the algorithm is as follow:

The elements of Formalized Analysis Model (FAM) in the form of different table schema are used as input to the first algorithm named *NodeGeneration* (devised in

section 4.3.1). The outputs of this algorithm are different BPMN nodes. This output along with the Array FAM_Flow is fed as input to the second algorithm named *FlowGeneration*. The Array FAM_Flow is formal method of storing the flow information of events of FAM. The *FlowGeneration* algorithm will generate the BPMN design elements.

4.4.1 Algorithm NodeGeneration to Generate BPMN Node

The algorithm *NodeGeneration* as proposed below will generate the BPMN nodes. We define the algorithm using the tuple relational calculus. The algorithm is proposed as follow.

The following query is the realization of rule 1 of section 4.3.It generates the *Start* of the #BPMN node. It selects the Graphical_notation from BPMN_node and map that with the #start event of the activity_node

$$\{t. \text{Graphical_notation} \mid \text{BPMN_node}(t) \wedge t.ID = '1' \wedge \\ \exists d (d.act_ID \mid \text{ActivityState}(d) \wedge d.activity_node = t.activity_node \wedge t.label = d.act_ID \wedge d.activity_node = 'start')\}$$

The following query is the realization of rule 2 of section 4.3.It generates the *End* of the # BPMN node. It selects the Graphical_notation from BPMN_node and map that with the #end event of the activity_node

$$\{t. \text{Graphical_notation} \mid \text{BPMN_node}(t) \wedge t.ID = '2' \wedge \\ \exists d (d.act_ID \mid \text{ActivityState}(d) \wedge d.activity_node = t.activity_node \wedge t.label = d.act_ID \wedge d.activity_node = 'end')\}$$

The following query is the realization of rule 3 of section 4.3.It generates the #*In-termediate* of the BPMN node. It selects the Graphical_notation from BPMN_node #and map that with the *action* or *decision* event of the activity_node

$$\{t. \text{Graphical_notation} \mid \text{BPMN_node}(t) \wedge t.ID = 3 \wedge \\ \exists d (d.act_ID \mid \text{ActivityState}(d) \wedge d.activity_node = t.activity_node \wedge t.label = d.act_ID \wedge (d.activity_node = 'action' \vee d.activity_node = 'decision'))\}$$

The following query is the realization of rule 4 of section 4.3.It generates the graphical #notation for *Parallel Gateway*. It selects the particular graphical notation and map this #with that activity_node of *ActivityState* where activity_node is 'fork' and the event_type #'of all the postElement of that activity node is basic'

$$\{t. \text{Graphical_notation} \mid \text{BPMN_node}(t) \wedge t.ID = 4 \wedge \\ \exists q (q.act_ID \mid \text{ActivityState}(q) \wedge q.activity_node = t.activity_node \wedge \\ t.label = q.act_ID \wedge q.activity_node = 'fork' \\ \exists r (r.postElement \mid \text{ActivityState}(r) \wedge r.act_ID = q.act_ID \wedge \\ \exists s (s.event_ID \mid \text{ActivityState}(s) \wedge s.act_ID = r.postElement \wedge \\ \exists p (p.event_ID \mid \text{UseCase}(p) \wedge p.event_ID = s.event_ID \wedge s.event_type = 'basic'))))\}$$

The following query is the realization of rule 5 of section 4.3.It generate the graphical #notation for *Exclusive-OR Gateway*. It selects the particular graphical notation

and map #this with that activity_node of *ActivityState* where activity_node is ‘fork’ and the #event_type of the one of postElement of that activity node is ‘basic’ and the event_type #of the other postElement of that activity node is ‘alternate’

$$\{t.Graphical_notation \mid BPMN_node(t) \wedge t.ID = 5 \wedge \exists q(q.act_ID \mid ActivityState(q) \wedge q.activity_node = t.activity_node \wedge t.label = q.act_ID \wedge q.activity_node = 'fork' \wedge \exists r(r.postElement \mid ActivityState(t) \wedge r.act_ID \neq q.act_ID \wedge \exists s(s.event_ID \mid ActivityState(s) \wedge s.act_ID = r.postElement \wedge \exists p(p.event_ID \mid Usecase(p) \wedge p.event_ID = s.event_ID \wedge s.event_type = 'basic' \vee s.event_type = 'alternate'))))\}$$

The algorithm *FlowGeneration* as proposed in section 4.3.2 will generate the flows between these nodes that are generated by the algorithm *NodeGeneration*.

4.4.2 Algorithm FlowGeneration to Generate the Flow between bpmn Nodes

We use an array representation *FAM_flow* to represent the flow between different activity nodes. *FAM_flow* is a part of our Formalized Analysis Model to depict the flow between different events of use cases of objects oriented systems.

The array *FAM_Flow* is an [n, 3] array where n is the number of flows in the formalized analysis model.

FAM_Flow[0][i] lists the source activity node of the flow for i=0 to n

FAM_Flow [1][i] lists the destination activity node of the flow i= 0 to n

FAM_Flow [2][i] lists the types of flow between A[0][i] and A[1][i] for i= 0 to n

Entries in *FAM_Flow* [2] [i] are of the following types



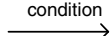

- 1) S indicates sequential flow
- 2) D indicates Default Flow
- 3) C indicates Conditional Flow
- 4) I indicate Iterative flow

Array FAM_Flow

A [] []	0	1	2	3	4	5	6	7	...	n-1
0(=ACi)	AC1	AC2	AC3	AC3	AC4	AC5				
1(=ACj)	AC2	AC3	AC4	AC5	AC6	AC2				
2(=value)	S	S	D	S	S	I				

Table *BPMN_Flow* stores different graphical notation of BPMN flow and are assigned with unique ID.

Table BPMN_Flow

ID	Name	Graphical_notation
1	Sequential Flow	
2	Default Flow	
3	Conditional Flow	
4	Iterative flow	

The algorithm *FlowGeneration* is proposed as follow:

Input:

Output of *Nodegeneration* algorithm, FAM_Flow[n,3] , Table BPMN_Flow

Algorithm:

```
for( m=0 ; m<=n-1;m++)
{
  flow.from = FAM_flow[m][0] ;
  flow.to = FAM_flow [m][1] ;
  If FAM_flow [m][2] = 'S'
  Flow.type = Select BPMN_Flow.Graphical_notation where BPMN_Flow.ID=1
  If FAM_flow [m][2] = 'D'
  Flow.type = Select BPMN_Flow.Graphical_notation where BPMN_Flow.ID=2
  If FAM_flow [m][2] = 'C'
  Flow.type = Select BPMN_Flow.Graphical_notation where BPMN_Flow.ID=3
  If FAM_flow [m][2] = 'I'
  Flow.type = Select BPMN_Flow.Graphical_notation where BPMN_Flow.ID=4
}
```

Output:

BPMN design elements

5 Case Study

Our proposed Automatic Transformation Model FAM2BP is explained with the help of the case study of a Banking System. We have taken four use cases where use case 2 (UC2) is the primary use case that includes use case 1(UC1) & use case 3(UC3) and is extended in specialized case like housing loan by use case 4(UC4). These use cases are tabulated in Table1 in the form of *Use case* Schema as defined in section 4.2. The events of individual use cases are stored in Table2 as *Event* schema which is defined in section 4.2. These events, which will be mapped with the *ActivityState* .The information regarding *ActivityState*, will be stored in Table 3 as *ActivityState* schema.

Table *ActivityState* contains the information regarding activity node. The entry in the table will have new activity like fork /join/decision etc apart from normal activity (start/action/end) which are mapped from the events of the *Event* table. The normal activity will carry the same event_ID as in the table *Event*. And new event_ID will be generated for the new activity. All the activities will be assigned an unique identifier.

In Figure 5, the different BPMN nodes and flows are generated using Tables 1, 2, 3, 4 and the array FAM_flow (defined in the previous section). The table 4 (Table ActUCRelation) which keeps information regarding any activity includes any use cases. Table 4 can be used for extend relation, as well. Here, UC4 extends UC2 and we can replace this with the relation UC4 includes UC2, which implies that UC4 has all the functionalities of UC2, along with its own functionalities. Henceforth, UC4 will have an activity which will include UC2. The *Reuse* field in table 3 is used to incorporate reusability (include, extend in terms of include relationship) of use cases. If the *Reuse* field is 'Y', then table 4 has to be checked to find which usecase has to be included by checking the UC_id field.

Table 1. Table Use Case

UC_id	UC_name
UC1	Verify Customer
UC2	Sanction Loan
UC3	Determine the maximum limit of loan amount
UC4	Sanction home Loan

Table 4. Table ActUCRelation

Act_ID	UC_id
AC10	UC1
AC11	UC3

Table 2. Table Event

UC_id	event_id	event_name	event_type
UC1	EV1	A customer has called the bank or visit the bank	Basic
UC1	EV2	The customer will be asked the requisite set of questions.	Basic
UC1	EV3	Customer is able answer all verification questions successfully	Basic
UC1	EV4	Customer is unable answer verification questions	Alternate
UC1	EV5	Verification is complete	Basic
UC2	EV1	A customer has called the bank	Basic
UC2	EV2	Includes UC1	Basic
UC2	EV3	Include UC3	Basic
UC2	EV4	Verify address	Basic
UC2	EV5	Finalization of interest rate	Basic
UC2	EV6	Calculation of EMI	Basic
UC2	EV7	Loan is sanctioned	Basic
UC3	EV1	Customer has applied for loan	Basic
UC3	EV2	Income and other factor are taken as input	Basic
UC3	EV3	The maximum loan limit of the customer is calculated	Basic
UC3	EV4	The maximum calculated limit is less than the requested loan limit.	Alternate
UC3	EV5	Customer loan amount is sanctioned	Basic
UC4	EV1	Customer has applied for home loan	Basic
UC4	EV2	Customer submit property details etc	Basic
UC4	EV3	The searching of property is done and searching result is satisfactory	Basic
UC4	EV4	The searching of property is done and searching result is not satisfactory	Alternate

Table *ActivityState* (table 3) contains the information regarding activity node. The entry in the table will have new activity like fork /join/decision etc apart from normal activity (start/action/end) which are mapped from the events of the *Event* table (table 2).

The normal activity will carry the same event_ID as in the table Event. And new event_ID will be generated for the new activity. All the activities will be assigned an unique identifier. As a result , the different BPMN nodes and flows are generated using Tables 1, 2, 3, 4 and the array FAM_flow (defined in the previous section). The Table ActUCRelation (table 4) which keeps information regarding any activity includes any use cases. Table 4 can be used for extend relation, as well. Here, UC4 extends UC2 and we can replace this with the relation UC4 includes UC2, which implies that UC4 has all the functionalities of UC2, along with its own functionalities. Henceforth, UC4 will have an activity which will include UC2.

Table 3. Table ActivityState

UC_id	event_ID	act_ID	activity_node	preElement	postElement	Reuse
UC1	EV1	AC1	Start	-----	AC2	N
UC1	EV2	AC2	action	AC1	AC3	N
UC1	F	AC3	fork	AC2	AC4 ,AC5	N
UC1	EV3	AC4	action	AC3	AC6	N
UC1	EV4	AC5	action	AC3	AC2	N
UC1	J	AC6	join	AC4 ,AC5	AC7	N
UC1	EV5	AC7	end	AC6	-----	N
UC2	EV1	AC8	start	-----	AC8	N
UC2	F	AC9	fork	AC8	AC10,AC11	N
UC2	EV2	AC10	action	AC9	AC12	Y
UC2	EV3	AC11	action	AC9	AC12	Y
UC2	J	AC12	join	AC10 ,AC11	AC13	N
UC2	EV4	AC13	action	AC12	AC14	N
UC2	EV5	AC14	action	AC13	AC15	N
UC2	EV6	AC15	action	AC14	AC16	N
UC2	EV7	AC16	end	AC15	-----	N

6 Conclusion

In this paper, we have proposed an approach for automated translation of Formalized Analysis Models, that consists of a formal grammar based description of UML models used in Analysis phase, to Business Processes. Design and development of software has become much more complex in the last decade, resulting in evolution of design and development paradigms. Object oriented systems have thus become an integral part of more complex Service Oriented Architecture (SOA). Evolution of software design and development from object oriented to SOA domain has become the necessity in this evolving scenario. This approach would help us in seamless evolution of object oriented systems to service oriented domain. As this model is based on a formal grammar, this approach can be automated resulting in correct and consistent transformations.

References

- [1] Cockburn, A.: Writing Effective Use Cases, 14th edn. Addison-Wesley, Reading (August 2005)
- [2] Dijkman, R., Dumas, M., Ouyang, C.: Semantics and Analysis of Business Process Models in BPMN. Information and Software Technology (IST) (2008)
- [3] Lübke, D.: Transformation of Use Cases to EPC Models. In: Proceedings of the EPK 2006 Workshop, Vienna, Austria (2006), <http://ftp.informatik.rwth-aachen.de/Publications/CEURWS/Vol-224/>
- [4] Object Management Group. Unified Modeling Language: Superstructure (2004), <http://www.omg.org/cgi-bin/doc?formal/05-07-04> (last access 2007-09-01)
- [5] Object Management Group. Business Process Modeling Notation (BPMN) 1.1 (January 2008)
- [6] Oestereich, B., Weiss, C., Schröder, C., Weikiens, T., Lenhard, A.: Objektorientierte Geschäftsprozessmodellierung mit der UML. d.punkt Verlag (2003)
- [7] Somé, S.: An approach for the synthesis of State transition graphs from Use Cases. In: Proceedings of the International Conference on Software Engineering Research and Practice, Las Vegas, Nevada, USA, June 23 - 26 (2003)
- [8] Somé, S.: Supporting Use Cases Based Requirements Simulation. In: Proceedings of the International Conference on Software Engineering and Practice (SERP 2004), Las Vegas, Nevada, USA, June 21-24 (2004)
- [9] Whittle, J.: A Formal Semantics of Use Case Charts, Technical Report ISE Dept, George Mason University, ISE-. TR-06-02, <http://www.ise.gmu.edu/techrep>
- [10] Lübke, D., Schneider, K., Weidlich, M.: Visualizing Use Case Sets as BPMN Processes. In: Requirements Engineering Visualization (REV 2008), Barcelona, Spain, September 8-12 (2008)
- [11] Chanda, J., Kanjilal, A., Sengupta, S., Bhattacharya, S.: Traceability of Requirements and Consistency Verification of UML Use case, Activity and Class Diagram: A Formal Approach. In: Proceedings of International Conference on Methods and Models in Computer Science (IEEE ICM2CS), New Delhi, December 14-15 (2009)

A Cross-Layer Framework for Adaptive Video Streaming over IEEE 802.11 Wireless Networks

Santhosha Rao¹, M. Vijaykumar¹, and Kumara Shama²

¹ Department of Information and Communication Technology

² Department of Electronics and Communication Engineering,
Manipal Institute of Technology, Manipal University, India
msanthosharao@gmail.com, vijayns02@gmail.com,
shama.kumar@manipal.edu

Abstract. Development in video compression techniques and wireless techniques has resulted in considerable amount of video streaming applications in the wireless networks. However, the time varying transmission characteristic of the wireless channels leads to poor performance of multimedia traffic over wireless networks. This results in longer packet delay, jitter and lower throughput that deteriorate the video quality considerably at the receiving end, thus diminishing the user experience. In this work, we propose cross-layer framework which optimizes the transcoding rate at the application layer depending upon the channel condition estimated using parameters associated with the data-link layer. Further, we evaluate the proposed frame work using NS-2 simulator with EvalVid framework. We use three different motion video sequences to evaluate the proposed frame work. Our simulation result shows that the proposed cross-layer frame work improves the video quality in all the three cases at the receiving end.

Keywords: adaptive video streaming, cross-layer, wireless networks.

1 Introduction

The use of the Wireless Local Area Networks in public areas, offices and homes has been scattering quickly and connection with internet is also increasing. Traffic carried by wireless networks is expected to be real time traffic such as video-on-demand, video conference and voice. To provide the quality video over wireless networks is challenging, due to the time-varying and erratic nature of a wireless channel and the strict delivery requirements of media traffic. Time varying transmission characteristic of the wireless channel leads to packet delay, packet loss, jitter and lower throughput. In video streaming, to set the play-out buffer at the receiving end is to reduce the jitter effect and smooth out the video played by the video client. However, additional packets are dropped by the play-out buffer because the important packet was lost or arrived later than allowable time. The traditional layered approach addresses this problem by implementing the prediction and adaptation algorithm at the lower layers of the protocol stack, specifically the transport, the medium access and the physical

layers. The lower layers are optimized independently without considering the effect on the other layers. These approaches do not adaptively support the real-time traffic such as the video streaming.

To increase the video quality at end user, we require an adaptive encoder, which considers the network conditions and varies its encoding rate accordingly. In order to achieve this, we require a feedback mechanism from lower layer which the existing layered architecture does not support. A cross-layer design is required to provide the necessary feedback from the lower layers. The concept of cross-layer design is based on inter-layer information exchange across the layers which aims at achieving mutual optimization of two or more layers. Even though this concept can be used in all communication networks, it is particularly appropriate in wireless networks because of the unique challenges concerned with the wireless environment. In this work, we use the layered architecture along with cross-layer information exchange to increase the video quality at end user.

Based on application requirements, different solutions to vary the video rate are proposed in literature. The rate-distortion optimized frame work is introduced in [1]. According to this framework, the packets are dropped intelligently based on their relative importance and priority. In certain applications the video encoder can adapt to the channel conditions by changing the compression degree and thus modifying the video rate [2]. The video encoder is able to get the channel quality by receiving the feedback information from the receiver side. However, such a proposal is hopeless when the round-trip delays are long. In [3] the authors pre-encoded video data at different rates. The end user selects the desired rate based on his requirement, which is not an appropriate approach for live video streaming. To mitigate the above mentioned problems we use the real-time transcoder for live video streaming in our approach.

We use retransmission information from data-link layer to estimate the network condition and vary the transcoder compression rate accordingly. In order to achieve this, we propose the cross-layer frame work which captures the retransmission information from data-link layer and feeds this information to the Application layer. The remaining part of the paper is organized as follows. Section 2 describes the proposed cross-layer frame work. Section 3 describes simulation set up and analysis of the result and section 4 provides a brief conclusion about our work.

2 The Proposed Cross Layer Framework

We considered IEEE 802.11 infrastructure mode wireless network and scenario where a video is streamed from a video server to base station through internet. The video from the video server is first received at the access point before being delivered to the appropriate wireless node. Therefore, the proposed cross layer framework is implemented in access point (AP). As shown in Fig.1 the frame work consists of the optimizer block along with the input/output interface to the protocol stack. We explain in detail, the optimizer block and its interfaces in the next subsections.

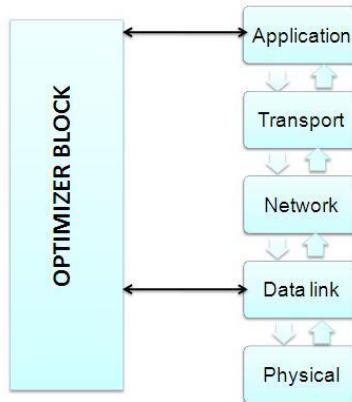


Fig. 1. Protocol stack for video streaming optimization

2.1 Input

The input to the optimizer block is obtained from the data-link layer. In data-link layer, information of packet transmission attempt is maintained in a counter. The data-link layer repeatedly transmits a packet until it receives a positive acknowledgment. Therefore, multiple attempts can be made to achieve successful transmission. Thus, the packet transmission attempts made at the data-link layer act as input data for the optimizer block.

2.2 Optimizer Block

Optimizer Block is similar to channel estimator. Optimizer will get the input from the data-link layer, based on this information it will estimate the channel condition. When input counter value is one, it means that there is no delay in network, thus indicating that the channel condition is good. When input counter value is two, it indicates that the channel condition is moderate. When input counter value is three or four, it indicates that the channel condition is bad. Based on the available channel information from the optimizer, the application layer then invokes the optimal approach to transcode the incoming video bit-stream.

2.3 Output

Optimizer block output is fed to the transcoder rate controller which will give the input parameter value to the transcoder. Based on this parameter, transcoder varies its encoding rate. When channel condition is good, it encodes the incoming video data at higher rate. When channel condition is moderate, it will encode the incoming video data at moderate rate and when channel condition is bad, it will encode the incoming video data at lower rate.

2.4 Video Transcoding Rate Calculation

Video compression is achieved by reducing the amount of data that is required to send. The main factor affecting the spatial detail is the quantization parameter. Details of the video are preserved when the Quantization Parameter (QP) is small. Some of the spatial detail is reduced when the QP is raised, leading to the decrease in bit rate. Hence, variation in the QP causes the variation in the bit rate. To control the transcoder mainly we require two parameters: i) target bit rate ii) QP. By default, transcoder recompresses the incoming frame at the rate of R (bits/frame). A transcoder is said to be an adaptable transcoder when it varies its transcoding rate according to the network conditions. When channel condition is good, we increase the transcoder target rate to $1.2R$ (R_{\max}) for higher video quality. When channel condition is moderate, transcoder target bit rate will be same as it is, to take the full advantage of the channel. When channel condition is bad, we decrease the transcoder target bit rate to $0.8R$ (R_{\min}), to reduce the retransmission, packet loss at end user, smooth-out the transcoded video stream and also reduce the load on the network. Here, we select the rate alteration factor of 1.2 because higher values lead to disorder of the pre-calculated bit-budget distribution in H.264 encoder [4], which should be avoided. Furthermore, in case of bad channel condition, we use the rate alteration of 0.8 because lower values change the video quality. To choose the QP when the target bit-rate is available, both the linear and quadratic rate quantization (RQ) models are suggested. The quadratic RQ model is not appropriate for real-time transcoding because of its complexity [5] in QP selection, even though the quadratic model has higher accuracy than the linear model. Accordingly, a linear RQ model [6] is chosen in this work to get a better balance in the tradeoff between accuracy and complexity for real-time video streaming application.

Summary: In Fig.2, flow chart represents the procedure used in the proposed cross layer framework. The first frame is fed to the transcoder and its target rate is initialized to R_{\max} . If the recent frame is not the first frame, then the transcoder computes the new transcoding QP value, based on the following information: available bandwidth, bit rate of the previous rate, frame rate and channel information from the optimizer block. If the incoming frame to transcoder is B frame, the calculated QP is used for transcoding. Otherwise, in case of I and P frames, previous calculated QP is used for transcoding. Optimizer block estimates the channel condition based on the number of transmission attempts for a packet. The channel condition information is fed back to the application layer for transcoding the next video frame.

3 Simulation Setup

We evaluate proposed cross layer framework using the NS-2 [8] network simulator, over IEEE 802.11 wireless network. CBR and FTP applications can be simulated using built in NS-2 agents, but there is no agent for video traffic application in NS-2. So, to achieve simulation of the video traffic, we use the EvalVid framework [9] and the JM reference software [7] to transcode the video sequences.

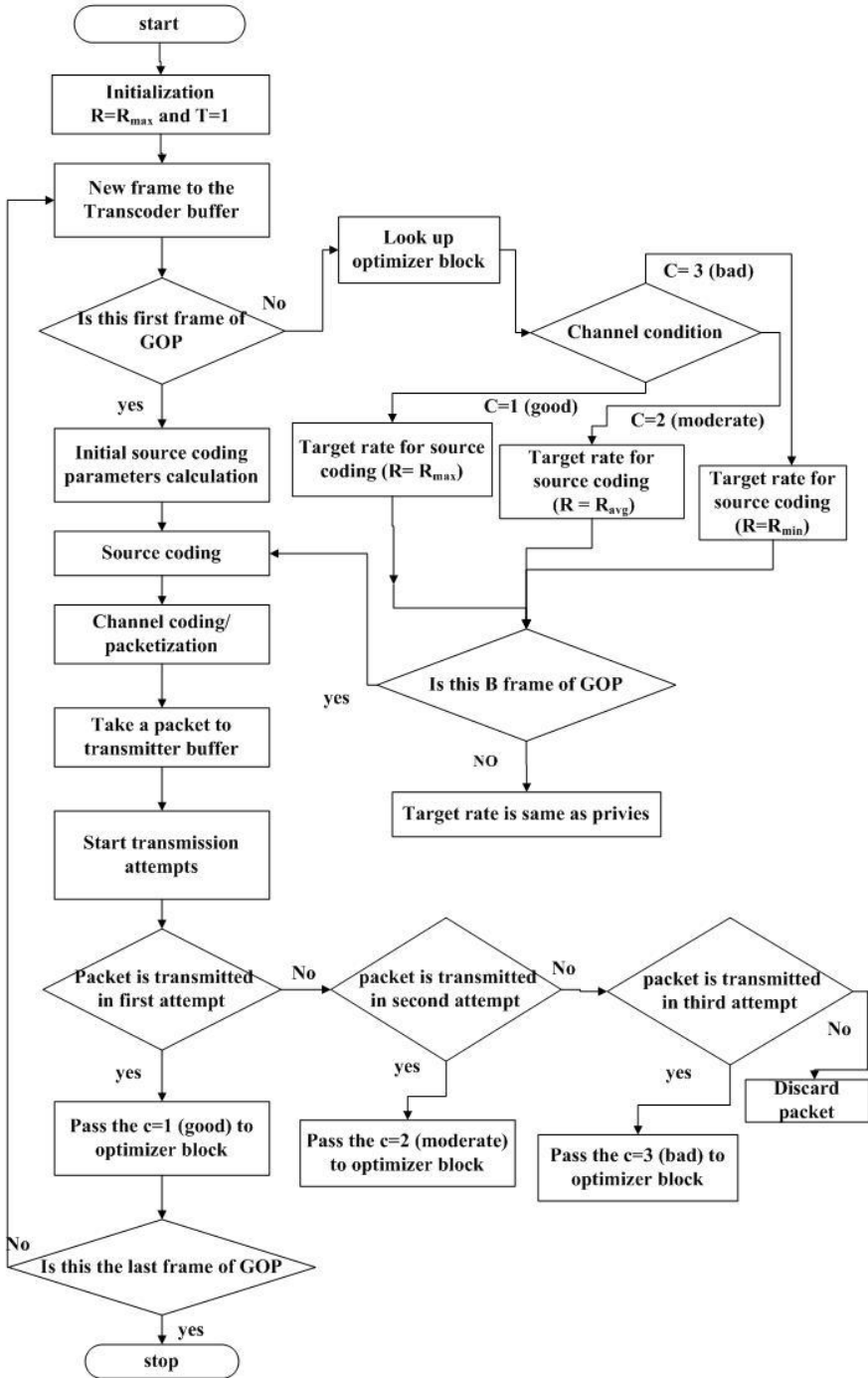


Fig. 2. Flow of operations in the proposed cross-layer framework

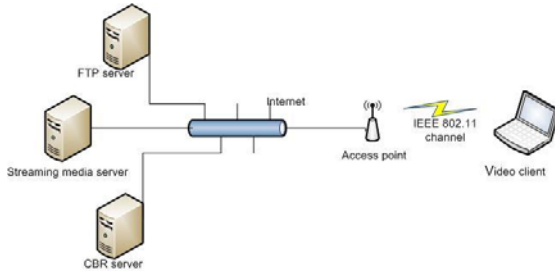


Fig. 3. Simulation model

To create realistic network conditions, we consider simulation environment as in Fig.2. The network traffic between the server and access point involves traffic due to FTP, CBR and the video data which is of our interest. FTP traffic is transmitted over TCP protocol and CBR traffic is transmitted over UDP protocol. The video traffic is transmitted over the myudp protocol. This protocol is similar to udp protocol. We use the myudp protocol to handle the video traffic in the simulation. The FTP source represents a bulk file transfer application with a maximum packet size of 1000 bytes and source rate is set at 512 kbps. The file considered is enough such that there is always data to transmit over the entire duration of the simulation. CBR represents the bursty traffic, with a maximum packet size of 1280 bytes and the source rate is set at 256 Kbps. In this simulation, we use mother-daughter, News and Foreman video sequences, which are example of the slow, medium and fast motion video clips respectively. These video clips are used as video traffic in simulation and the first 250 frames of each of these video clips are encoded using H.264 encoder. The first frame of the video sequence is an I-frame, while the subsequent frames are P-frames and B-frames. The video frame rate is set to 30 frames per second. Context Adaptive Binary Arithmetic Coding (CABAC) [4] was used for the entropy encoding and RD optimization [4] was enabled. Each video frame is segmented into small packets in transmission and the maximum packet size was set to 1000 bytes. The link between the base station and the video receiver is IEEE 802.11b 2Mbps link. For simplicity, we assume that the link between the base station and the video server has 10Mbps bandwidth and 10 ms latency. The Simulation results are analyzed in next section.

3.1 Results

Objective quality: Peak-signal-to-noise-ratio (PSNR) is one of the most well known objective metrics to evaluate the application level Quality of Services (QoS) of video transmissions and is therefore used in this work to measure video quality. PSNR measures the error between the encoded video at sender side and decoded video at receiver side. Fig.4, Fig.5 and Fig.6 show the improvement in PSNR with the proposed cross-layer frame work for the test video sequences Mother-Daughter, News, and Foreman respectively. Two different scenarios are considered to compare the video quality at end user. (1) Application layer parameters is adapted considering the channel conditions (denoted by with cross layer), (2) without any adaptation of

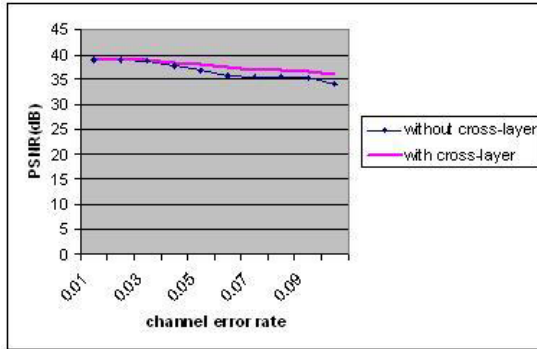


Fig. 4. Objective video quality of the Mother-daughter test video sequence

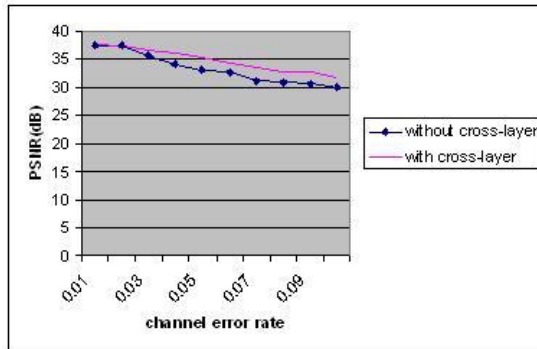


Fig. 5. Objective video quality of the News test video sequence

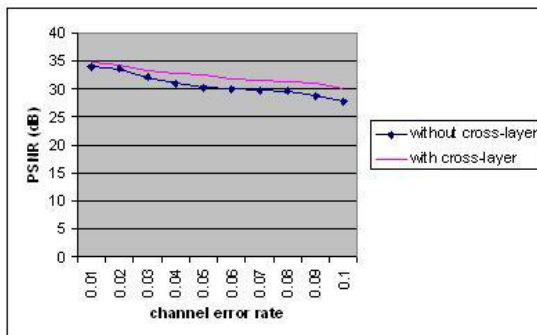


Fig. 6. Objective video quality of the Foreman test video sequence

parameters at the application layer (denoted by without cross-layer). In the entire three test video sequences, when channel state is good there is no tangible improvement in video quality, in case of proposed framework. When channel condition is moderate, the proposed cross-layer framework performs nearly 1 dB better in terms of PSNR than the case when no such proposed frame work is used. When channel condition is bad, the proposed cross-layer framework performs 1-2 dB better in terms of PSNR than the case when no such cross-layer approach is used. This improvement is more in case of medium and fast motion test video sequences than slow motion test video sequences. The improvement in the video quality in case of moderate and bad channel conditions is due to the variation of video rate at the application layer.



Fig. 7. Subjective quality test for three test video sequences

Subjective quality: We also evaluate the subjective quality of three test video sequences and results are as shown in Fig.4. The source format is cif (common intermediate format), where the pixel values are 352×288 . The Fig.7(a), Fig.7(a') and Fig.7(a'') are the reference video frames, and are given here for the comparison purpose. Fig.7(b),

Fig.7(b') and Fig.7(b'') are the screenshots of the three test video clips when they are reconstructed at the decoder with the channel error rate of 3 percentage, and in the absence of the proposed framework. The degradation in video quality is clearly visible at the central bottom part of Fig.7(b) , lower part of Fig.7(b') and at lower part of Fig.7(b''). This shows that the viewers will experience a poor quality, while watching the video. When the proposed cross-layer based framework was used for video streaming, the results are as shown in the Fig.7(c), Fig.7(c') and Fig.7(c''). This shows considerable improvement in the subjective quality of the video. The corresponding PNSR values and the gain in PSNR with the proposed frame work are listed in Table 1.

Table 1. PSNR values for the subjective quality test

	Mother – daughter	News	Foreman
Original (dB)	39.23	38.5	36.24
without cross-layer (dB)	36.64	33.30	30.30
with cross-layer (dB)	37.70	34.90	32.40
with versus without CL Gain (dB)	1.06	1.60	2.1

4 Conclusion and Future Work

In this paper, we have proposed a cross-layer framework for adaptive video streaming in wireless networks. The proposed cross-layer framework optimizes the transcoding rate at application layer depending upon the channel condition estimated using parameters associated with the data-link layer. We have evaluated the proposed cross-layer framework using NS-2 simulator with EvalVid framework for slow, medium and fast motion video sequences. When channel condition is moderate, the proposed cross-layer framework performs nearly 1 dB better in terms of PSNR than the case when no such proposed framework is used. When channel condition is bad, the proposed cross-layer framework performs 1-2 dB better in terms of PSNR than the case when no such cross-layer approach is used. Future work will be to implement the proposed framework in the real network setup.

References

1. Shen, H., Sun, X., Wu, F.: Fast H.264/MPEG-4 AVC transcoding using power- spectrum based rate-distortion optimization. *IEEE Transactions on Circuits and Systems for Video Technology* 18(6), 746–755 (2008)
2. Bolot, J.-C., Turetli, T.: Experience with control mechanisms for packet video in the internet. *Computer Communication Review* (1998)
3. Ozcelebi, T., Civanlar, M.R., Tekalp, A.M.: Minimum delay content adaptive video streaming over variable bitrate channels with a novel stream switching solution. In: *Proc. IEEE International Conference on Image Processing (ICIP 2005)*, vol. 1, pp. 209–212 (2005)
4. ITU-T Rec. H.274 and ISO/IEC 14497-10 (MPEG4-AVC), Advanced video coding for generic audiovisual services, ver. 1, May 2003; ver. 2, January 2004; ver. 3 (withFRExt), September 2004; ver. 4. July 2005

5. Jiang, M., Ling, N.: Low-delay rate control for real-time H.264/AVC video coding. *IEEE Transactions on Multimedia* 8(3), 467–477 (2006)
6. Li, Z.G., Pan, F., Lim, K.P., Feng, G.N., Lin, X., Rahardaj, S.: Adaptive basic unit layer rate control for JVT. In: *Proceedings of the JVT-G012, 7th Meeting, Pattaya, Thailand (March 2003)*
7. JM reference software, Maintained by K. Suhring (June 2008), <http://iphome.hhi.de/suehring/tml/>
8. Network Simulator, <http://www.isi.edu/nsnam/ns/>
9. EvalVid-A video quality evaluation tool-set, <http://www.tkn.tuberlin.de/research/evalvid/>

FIFO Optimization for Energy-Performance Trade-off in Mesh-of-Tree Based Network-on-Chip*

Santanu Kundu, T.V. Ramaswamy, and Santanu Chattopadhyay

Department of Electronics and Electrical Communication Engineering,
Indian Institute of Technology, Kharagpur, Kharagpur – 721302, India
skundu@ece.iitkgp.ernet.in, elegantfrmsrkr@gmail.com,
santanu@ece.iitkgp.ernet.in

Abstract. This paper presents an exhaustive study about the impact of FIFO optimization on performance and energy consumption in a Mesh-of-Tree (MoT) based Network-on-Chip (NoC) architecture. A generic NoC router has FIFO at each input and output channel. The paper shows that FIFO is the most energy hungry component in a NoC router. On the design trade-off front, we establish that elimination of FIFO from the output channel reduces energy consumption significantly at the cost of marginal performance degradation.

Keywords: network-on-chip, mesh-of-tree, performance and cost evaluation, FIFO optimization, energy-performance trade-off.

1 Introduction

Network-on-Chip (NoC) is a new paradigm for designing future many core based SoCs [1] where IP cores communicate with each other with the help of on-chip routers and point-to-point interconnection links. Routers are the main components of interconnection architecture for routing information from source to destination cores. The router has a functionality to store the incoming information and transfer that to any one of its output ports. A generic NoC router has FIFO buffer at every input port to receive incoming data and at every output port to store those data before they are put on the output port [2]. A crossbar switch helps to connect an input buffer to any of the output buffers. There are considerable amount of works that uses both input and output buffering in a router [3–5]. Researchers have also proposed the router architecture having the FIFO buffer only at the input port [6–8]. While [9] reports the buffer placement and sizing to be a very important issue in NoC design, we could not locate any study that clearly proves superiority of one over the other. This motivates us to find out the impact of FIFO on energy and performance.

For NoC, topologies like mesh [10], torus [11], folded torus [12], fully binary tree [13], fat-tree [14], octagon [15], butterfly fat tree (BFT) [5], and Mesh-of-Tree (MoT) [16] have already been proposed. A detailed comparative performance evaluation of a set of recently proposed NoC architectures with Poisson and self-similar traffic models

* This work is partially supported by Department of Science and Technology (DST), Govt. of India (SR/S3/EECE/0012/2009), Dt. 20th May, 2009.

have been performed in [17]. The authors of [18] have designed MoT based wormhole router architecture to show that MoT performs better than mesh based NoC under Poisson distributed traffic. Although MoT is a potential candidate in interconnection network, its energy consumption profile and area overhead in NoC paradigm is unexplored till date. This also motivates us to perform the energy, area, and performance evaluation by varying the FIFO position and FIFO depth on MoT based network. The contributions of this paper are as follow:

1. A detailed performance and energy consumption of MoT based network has been evaluated.
2. The trade-off between energy and performance has been carried out by varying FIFO depth and position of MoT based network. The experimental results show that elimination of output buffer reduces the energy consumption significantly at the cost marginal performance degradation.
3. The area requirement of MoT based network has been investigated. The area overhead of MoT based networks is quite small compared to the total SoC area.

The rest of this paper is organized as follows: Section 2 describes the methodology for performance evaluation of NoC and associated cost metrics. The experimental results and analysis have been described in Section 3. Section 4 depicts the FIFO optimization for energy-performance trade-off in MoT based network and also focuses the trend of similar study in mesh and BFT based network. The area requirement of MoT based network for different FIFO placement and sizing has also been shown in this section. Section 5 concludes the present work.

2 Methodology for Performance Evaluation and Cost Metrics

For evaluating performance of the interconnection network, we have designed a cycle-accurate NoC simulator (in SystemC). The simulator operates at the granularity of individual architectural components of wormhole router. The current version of our simulator supports MoT based network having 32 IP cores where each IP core is taken as a square of side 2.5 mm as considered in [19].

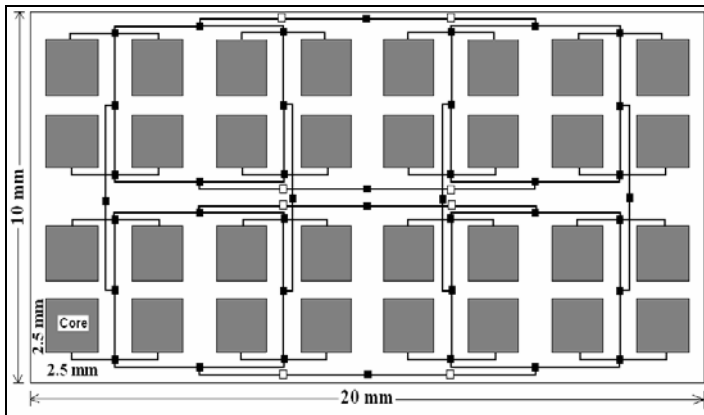


Fig. 1. Hand layout of MoT based NoC having 32 cores

Fig. 1 shows a hand layout of 32 core based NoC mapped onto $20\text{ mm} \times 10\text{ mm}$ silicon die in 4×4 MoT fashion where 2 layers, horizontal and vertical, are needed for routing. In any tree based topology link length increases with growing network size. As wire delay increases exponentially with its length, in MoT, we have pipelined the links whose length is more than 2.5 mm . The register used for pipelining is shown as small white node, whereas, the routers are denoted as small black nodes in Fig. 1. Thus the length of inter router links and the core to router links are 2.5 mm and 1.25 mm respectively, assuming the inter core spacing is negligible.

Table 1 shows the number of each type of links in MoT network. The router architecture has been implemented in Verilog HDL and synthesized in *Synopsys Design Vision* in 90 nm technology and the critical path delay is found to be 600 ps . In this paper, we have applied mesochronous clocking strategy to all the routers having same frequency (1.66 GHz) with varying phase. The cores are modeled via traffic generators and traffic receptors. Each traffic generator module generates self-similar traffic [20] as it shows the actual burstiness of the network traffic. In this simulator, the user may also choose between uniform and localized traffic patterns for the packets. Locality factor is denoted by LF and its value is zero for uniformly distributed traffic. It is defined as the ratio of local traffic to total traffic. If $LF = 0.5$, then 50% of the traffic will go to the nearest cluster from source (S). The rest of the traffic will be distributed according to their distances from the source such that a destination at nearer cluster will get more traffic compared to one at farther cluster. Now, if there is more than one destination in a cluster, the traffic will be randomly distributed among them. In our simulation, we have fixed the packet length to be 64 flits with flit size of 32 bits. The packet injection is continued throughout the entire simulation time of $200,000$ cycles of the routers' clock including $10,000$ warm-up cycles to make the network stable from initial transient effects. For accurate estimation of the energy consumption by the network, we have assumed that each traffic generator module injects the traffic into the network with a switching activity factor of 0.9 as that is expected to introduce a large number of transitions, and thus power consumption in the network.

Table 1. Estimation of link length in MoT topology connecting 32 cores

Topology	Number of Links	
	1.25 mm	2.5 mm
MoT	64	112

2.1 Performance Metrics

The performance of an on-chip communication network is characterized by its throughput and average overall latency. Throughput (TP) is the maximum accepted traffic from the network and it is related to the peak data rate sustainable by the network. It is defined as [18],

$$TP = (Total\ Packets\ Received * Packet\ Length) / (Number\ of\ IP\ blocks * Total\ time)$$

Total Packets Received refers to the number of packets that successfully arrive at their destination IPs, *Packet Length* is measured in terms of flits, *Number of IP blocks* is the number of functional IP blocks involved in the communication, and *Total time*

denotes the simulation time (in clock cycles). Network *bandwidth* refers to the maximum number of bits that can be sent successfully to the destination through the network per unit time. It is represented as bits/sec (*bps*).

$$BW = (\text{Throughput} * \text{Number of IP cores attached} * \text{flit size}) / (\text{clock period})$$

Depending on the source/destination pair and the routing algorithm, each packet may have a different latency. There is also some overhead in the source and destination that contributes to the overall latency. So, for a given message i , the overall latency (L_i) is defined as,

$$L_i = \text{sender overhead} + \text{transport latency} + \text{receiver overhead}$$

Let P be the total number of packets reaching their destination IPs, the average overall latency, L_{avg} , is then calculated as:

$$L_{avg} = \sum_{i=1}^P L_i$$

2.2 Cost Metrics

Energy consumption: Energy consumption of each router is determined by using *Synopsys Prime Power* in *90nm* CMOS technology by running their gate level netlist. We have calculated the number of toggles of every individual I/O pin of the router and their probability of remaining in *logic-1* state for the entire simulation time from our NoC simulator and fed this information to *Prime Power*. We have fixed the clock frequency at *1.66 GHz* and simulated it with the following parameters: Process = 1, Supply Voltage = 1V, and Temperature = 75°C .

NoC links have been characterized separately from that of router and have been modeled as semi-global interconnect. Copper wire (resistivity = $17 \text{ n}\Omega\text{-m}$) of metal layer 4 has been chosen as interconnection link. The width and thickness of the wire have been taken to be $0.25 \mu\text{m}$ and $0.5 \mu\text{m}$ respectively. The spacing between two adjacent wires is kept at $0.25 \mu\text{m}$. The spacing between two adjacent metal layers is fixed at $0.75 \mu\text{m}$ and is filled by a dielectric material having relative permittivity of 2.9. Fig. 2 shows the cross-section view of interconnect wires sandwiched between two orthogonal metal layers carrying the ground signals. The parasitic resistance, capacitance, and inductance have been extracted by the *Field-Solver* tool from *HSPICE* supporting *90 nm* technology with a three wire model.

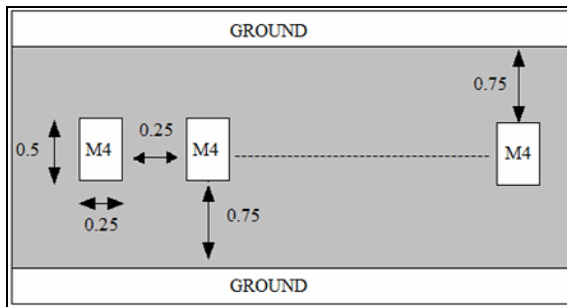


Fig. 2. Cross-section view of interconnect wires

Energy consumption of the links is determined separately from that of the routers. Energy of a line for different transition can be calculated using the three wire model with the assumption that the coupling effect on the middle line by non adjacent lines is negligible. The input signal is a non ideal signal with rise and fall times of 80 ps and the load capacitance at the other end is 5 fF . Table 2 shows the information about inverter sizing (multiple of minimum sized inverter) and line delay including driver, repeater, and receiver for individual links. Energy consumption of the middle wire has been extracted for all possible 64 transitions on the three wires. This information is used to calculate link energy from the network simulator for $200,000$ clock cycles.

Table 2. Buffer sizing and worst case delay in three wire model

Line Length	Driver Size	Receiver Size	No. of Repeater	Repeater Size	Link Delay (ps)
1.25 mm	30X	20X	-	-	427
2.5 mm	50 X	-	1	100 X	430

Area requirement: To estimate the silicon area occupied by each router, we have developed their Verilog models and synthesized using *Synopsys Design Vision* in 90 nm CMOS technology supporting *Faraday* library to generate gate-level netlist. The synthesis tool reports the silicon area occupied by the design in μm^2 . The inter-router link area estimation has been performed by determining its length from the hand layout of SoC. The width and inter-wire spacing of the links is taken from the interconnect dimension as shown in Fig. 2.

3 Experimental Results and Analysis

We have applied the above mentioned evaluation methodology to find out the performance and energy consumption of MoT based network consisting of 32 IP cores by varying offered load in self-similar traffic. FIFO of depth 6 is used at both input and output channels to implement the wormhole router.

3.1 Accepted Traffic vs. Offered Load

The accepted traffic depends on the rate at which the IP blocks inject traffic into the network. Ideally, accepted traffic should increase linearly with increasing load at a slope of 45° . However, due to the limitation of routing and arbitration strategy and unavailability of enough buffer space within the wormhole router (FIFO depth \ll size of the packet), the network suffers from contention. Therefore, the accepted traffic saturates after a certain value of the offered load. Fig. 3a depicts this scenario for uniformly distributed traffic in MoT based network. The maximum accepted traffic where the network is saturated is termed as throughput as defined above and it relates to the maximum sustainable data rate by the network.

3.2 Throughput vs. Locality Factor

The effect of traffic spatial localization on network throughput is shown in Fig. 3b. As the locality factor increases, more number of traffic are destined to their local clusters, thus traversing lesser number of hops which in turn increases throughput. For 4×4 MoT, the possible distances (d) of the destinations from any source are $d = 0, 2, 4, 6,$ and 8 . There is only one destination core at the nearest cluster. Moreover, the node degree is less in MoT based network and hence the contention is also less. This explains the increased throughput in MoT with the increase in locality factor.

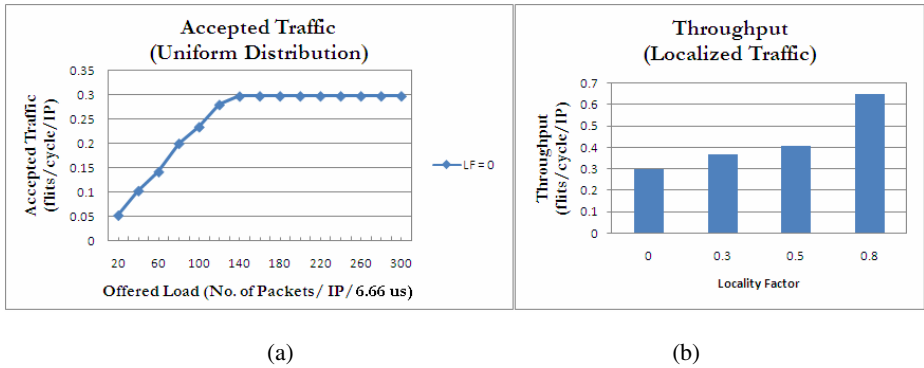


Fig. 3. Performance variation profile. (a) Variation of accepted traffic with offered load with uniform traffic, (b) Variation of Throughput with Locality Factor

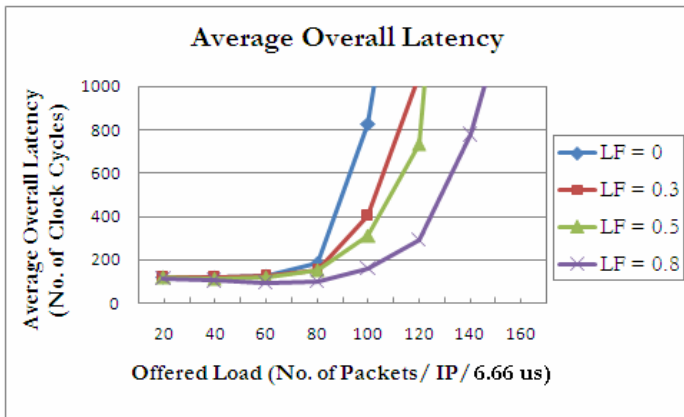


Fig. 4. Latency variation profile by varying offered load and locality factor

3.3 Average Overall Latency at Different Locality Factors

The average overall latency of any network depends on both the offered load and the locality factor. Fig. 4 shows the average overall latency profile for MoT by varying the offered load and locality factor. It shows that at lower traffic, the latency variation

is not significant. This is due to the fact that at lower traffic, the contention in the network is less, but it will increase as the offered load increases, which in turn increases the latency. The simulation result shows that as the offered load increases towards the network saturation point, the latency increases exponentially which signifies that the packets will take much longer time to reach their destinations. So, it is always desirable to operate the network below its saturation point. The effect of spatial localization of traffic on average overall latency is also studied and is shown in Fig. 4. It has been observed that localization of traffic has significant impact on MoT topology as the latency decreases with increase of locality factor. As the locality factor increases, more traffic will go to local cluster, hence less contention in the network. Therefore, the network will be able to carry more traffic before going to saturation, which in turn enhances the operating point of the network.

3.4 Energy Consumption at Different Locality Factors

Total energy consumption in NoC is the summation of energy consumed by the routers and inter-router communication links. Both these factors are network topology dependent. The total energy consumption of MoT based network for uniformly distributed self-similar traffic has been shown in Fig. 5 (simulation for 200,000 clock cycles with 600 ps clock period has been taken as evaluation parameter).

Fig. 5a shows that the energy consumption increases linearly with the offered load but saturates as the offered load increases to the throughput limit. Beyond saturation, no additional packets can be injected successfully into the network and, consequently, no additional energy is consumed. Fig. 5b depicts the component wise energy consumption of the network at saturation. We have observed that the energy consumption of all the FIFOs is 52% of the total network energy consumption, whereas, all the inter-router links consume only 38% of it. The combined energy consumption of rest of the router components is 10% of the same.

The average energy consumption per cycle of MoT based NoC at saturation with uniformly distributed and localized traffic has been shown in Fig. 6a. It can be noticed that energy consumption decreases as the locality factor increases. This happens due

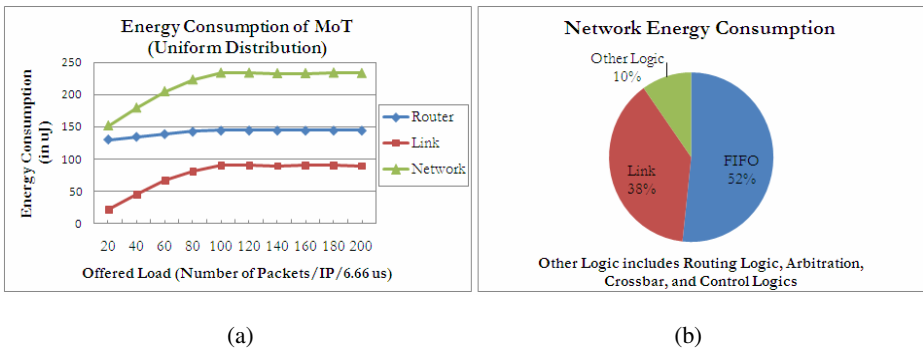


Fig. 5. Total Energy consumption profile in MoT with offered load at uniformly distributed traffic. (a) Network Energy consumed by all Routers and Links, (b) Energy consumed in percentage by FIFOs, wires, and other logics at saturation.

to the fact that as the locality factor increases, packets will traverse fewer hops and hence consume lesser switching energy. As FIFO is the most energy consuming component of the network, energy consumption varies with the total number of FIFO used to build the network, or in other words, with the topology. To get an idea about energy spent per packet, we compute the average packet energy. This is another important attribute for characterizing NoC structures. Fig. 6b shows the packet energy consumed at different locality factors at saturation. As the energy consumption profile decreases and the number of accepted traffic increases with increase in locality factor, packet energy consumption also decreases with increase in locality factor.

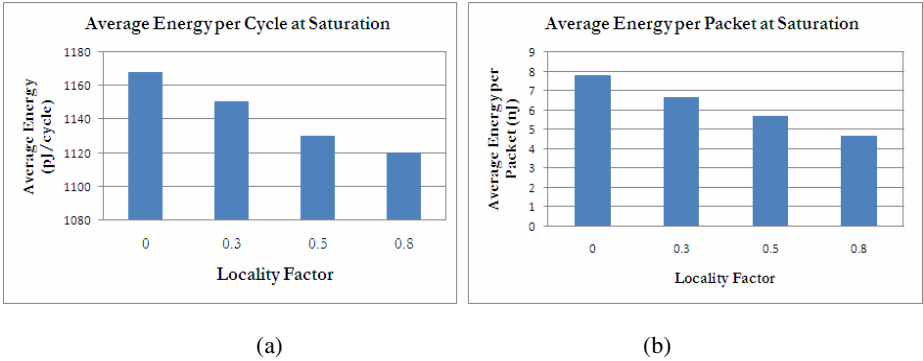


Fig. 6. Average Energy consumption at network saturation for different Locality Factors (a) per Cycle Energy, (b) per Packet Energy

4 FIFO Optimization for Energy and Performance Trade-off

In the previous section, it has been observed that FIFO is the most energy consuming element in the network. When contention occurs, the input channel FIFO with higher depth allows more data flits to make progress until an output channel is available. On the other hand, the output channel FIFO with higher depth allows more flits to cross the crossbar when the FIFO of the next router's input channel is full for some cycles. Decreasing the depth of any of the FIFOs will generate full signal frequently and thus the communication affected and hence a negative impact on performance. In this section, we are concentrating on optimizing the FIFO depth and FIFO position for energy and performance trade-off in MoT based network. In our experiment, we have evaluated the throughput, latency, and energy consumption of MoT based network by varying the depth of FIFO in both input and output channels. We have also experimented by eliminating FIFO from the output channel. Table 3 reports the % performance degradation and network energy saving in MoT at saturation for different FIFO size and position at different locality factors. *FIFO Depth i - j* in 2^{nd} column of Table 3 signifies that FIFO at input channel has depth i and that at output channel has depth j ($j=0$ signifies no FIFO at output channel). In this study, we have taken *FIFO Depth 6-6* (both input and output channel FIFO depths are 6) as the reference.

Table 3 shows that elimination of output channel FIFO in MoT router has very little impact on throughput and latency. As the FIFO depth of all the input channels is 6,

after getting the grant signal from any output channel, header will pass to the next router by saving the latency of output channel FIFO. This is diminishing the negative impact of eliminating the output channel FIFO to some extent. Network energy consumption also decreases with reduction of total FIFO size of the router. From Table 3, it can be concluded that elimination of output buffer reduces the energy consumption significantly at the cost marginal performance degradation.

Table 3. Energy Performance Trade-off in MoT at saturation (load = 300)

	FIFO Depth	Locality Factor			
		0	0.3	0.5	0.8
% Throughput Degradation	6-6	-	-	-	-
	4-6	7.183	5.763	4.574	3.504
	4-4	15.024	11.318	7.944	6.261
	6-0	1.192	1.391	0.229	0.156
	4-0	12.674	8.642	7.029	6.439
% Latency Increment	6-6	-	-	-	-
	4-6	44.94	58.539	53.831	102.942
	4-4	62.889	107.06	103.456	146.595
	6-0	13.641	12.187	11.716	14.317
	4-0	71.101	130.541	131.809	163.581
% Energy Saving	6-6	-	-	-	-
	4-6	14.425	13.623	12.403	13.668
	4-4	23.93	23.206	21.516	22.235
	6-0	36.638	36.899	36.891	36.443
	4-0	41.043	42.06	41.228	41.779

The final cost metric considered in this study is the overall area requirement to instantiate the MoT based networks. Fig. 7 shows the area requirement for varying FIFO depth and position, in terms of routers and inter-router links. It is observed that

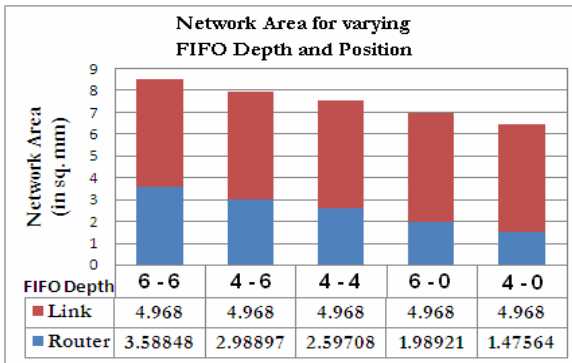


Fig. 7. Area occupied by the routers and wires to form the network of 32 cores having 2.5 mm × 2.5 mm dimension by varying FIFO depth

the total area occupied by the MoT based network with FIFO Depth 6-6 is *4.10* percent of the total SoC area where each core area is considered to be 6.25 sq. mm. Reduction of FIFO depth or elimination of FIFO from the output channel will reduce the overall area requirement. Therefore, the area overhead of the network is quite small compared to the total SoC area.

5 Conclusion

In this work we have explored the performance evaluation and energy consumption of MoT based generic NoC. We have shown that FIFO placement and sizing play crucial roles in determining the performance and energy consumption of NoC. From the experimental results, it can be concluded that instead of reducing the depth of the FIFO, eliminating it completely from the output channel reduces energy consumption significantly at the cost of marginal performance degradation. Moreover, it also reduces the overall area requirement.

References

1. Benini, L., Micheli, G.D.: Network on chips: A new SoC paradigm. *IEEE Computer* 35(1), 70–78 (2002)
2. Jantsch, A., Tenhunen, H.: *Networks on chips*, pp. 89–90. Kluwer Academic Publishers, Dordrecht (2003)
3. Zhang, Y.P., Jeong, T., Chen, F., Wu, H.: A study of the on-chip interconnection network for the IBM Cyclops64 multi-core architecture. In: *Proc. of IEEE International Parallel and Distributed Symposium, IPDPS* (2006)
4. Wentzlaff, D., Griffin, P., Hoffmann, H., Bao, L., Edwards, B., Ramey, C., Mattina, M., Miao, C.C., Brown, J.F., Agarwal, A.: On-chip interconnection architecture of the TILE processor. *IEEE Micro* 27(5), 15–31 (2007)
5. Pande, P.P., Grecu, C., Ivanov, A., Saleh, R.: High-throughput switch-based interconnect for future SoCs. In: *Proc. of IEEE Int'l Workshop on System-on-Chip for Real-Time Applications*, pp. 304–310 (2003)
6. Kumar, A., Kundu, P., Singh, A.P., Peh, L.S., Jha, N.K.: A 4.6Tbits/s 3.6GHz single-cycle NoC router with a novel switch allocator in 65nm CMOS. In: *Proc. of IEEE International Conference on Computer Design (ICCD)*, pp. 63–70 (2007)
7. Zeferino, C., Susin, A.: SoCIN: a parametric and scalable network on chip. In: *Proc. of Symp. on Integrated Circuits and System Design (SBCCI)*, pp. 169–174 (2003)
8. Kavaldjiev, N., Smit, G.J.M., Jansen, P.G.: A Virtual Channel Router for On-Chip Networks. In: *Proc. of IEEE Int'l SOC Conference*, pp. 289–293. IEEE Computer Society Press, Los Alamitos (2004)
9. Ogras, U.Y., Hu, J., Marculescu, R.: Key Research Problems in NoC Design: A Holistic Perspective. In: *Proc. of the IEEE/ACM Int'l Conf. on HW-SW Codesign and System Synthesis* (2005)
10. Kumar, S., Jantsch, A., Soininen, J.P., Forsell, M., Millberg, M., Oberg, J., Tiensyrja, K., Hemani, A.: A Network on Chip Architecture and Design Methodology. In: *Proc. of ISVLSI*, pp. 117–124 (2002)
11. Dally, W.J., Towles, B.: Route Packets, Not Wires: On-Chip Interconnection Networks. In: *Proc. of DAC*, pp. 684–689 (2001)

12. Dally, W.J., Seitz, C.L.: The Torus Routing Chip. *Journal of Distributed Computing* 1(4), 187–196 (1986)
13. Jeang, Y.L., Huang, W.H., Fang, W.F.: A Binary Tree Architecture for Application Specific Network on Chip (ASNOC) Design. In: *IEEE Asia-Pacific Conf. on Circuits and Systems*, pp. 877–880 (2004)
14. Guerrier, P., Greiner, A.: A Generic Architecture for on-chip packet-switched Interconnections. In: *Proc. of DATE*, pp. 250–256 (2000)
15. Karim, F., Nguyen, A., Dey, S.: An Interconnect Architecture for Networking Systems on Chips. *IEEE Micro* 22(5), 36–45 (2002)
16. Kundu, S., Chattopadhyay, S.: ‘Mesh-of-Tree Deterministic Routing for Network-on-Chip Architecture. In: *ACM Great Lake Symposium on VLSI (GLSVLSI)*, pp. 343–346 (2008)
17. Pande, P.P., Grecu, C., Jones, M., Ivanov, A., Saleh, R.: Performance Evaluation and Design Trade-Offs for MP-SOC Interconnect Architectures. *IEEE Trans. on Computers* 54(8), 1025–1040 (2005)
18. Kundu, S., Chattopadhyay, S.: Network-on-chip Architecture Design based on Mesh-of-Tree Deterministic Routing Topology. *Int’l Journal of High Performance Systems Architecture* 1(3), 163–182 (2008)
19. Feero, B.S., Pande, P.P.: Networks-on-Chip in a Three-Dimensional Environment: A Performance Evaluation. *IEEE Trans. on Computers* 58(1) (2009)
20. Park, K., Willinger, W.: *Self-Similar Network Traffic and Performance Evaluation*. A Wiley-Interscience Publication, John Wiley & Sons, Inc., (2000)

Outliers Detection as Network Intrusion Detection System Using Multi Layered Framework

Nagaraju Devarakonda¹, Srinivasulu Pamidi², Valli Kumari V.³, and Govardhan A.⁴

¹ Department of Computer Science & Engg., Acharya Nagarjuna University, Nagarjuna Nagar

² Department of Computer Science & Engg., V.R. Siddhartha Engineering College, Vijayawada

³ Department of CS &SE, A.U. College of Engineering, Andhra University, Visakhapatnam

⁴ Principal, JNTUH College of Engineering, Jagityala, Karimnagar (District), A.P.

dnagaraj_dnr@yahoo.co.in, srinivasulupamidi@yahoo.co.in,

vallikumari@gmail.com, govardhan_cse@yahoo.co.in

Abstract. Outlier detection is a popular technique that can be utilized for finding Intruders. Security is becoming a critical part of organizational information systems. *Network Intrusion Detection System (NIDS)* is an important detection system that is used as a counter measure to preserve data integrity and system availability from attacks [2]. However, current researches find that it is extremely difficult to find out outliers directly from high dimensional datasets. In our work we used entropy method for reducing high dimensionality to lower dimensionality, where the processing time can be saved without compromising the efficiency. Here we proposed a framework for finding outliers from high dimensional dataset and also presented the results. We implemented our proposed method on standard dataset kddcup'99 and the results shown with the high accuracy.

Keywords: Outliers, Intruders, classification, z-score, distance-based, accuracy.

1 Introduction

Computer based Information Systems are becoming an integral part of our organizations. An Information System is a computerized system which contains organization information which serves the organization in its various activities and functions. Computer Security is the ability to protect a computer system and its resources with respect to confidentiality, integrity, and availability. Various protocols, firewalls are in existence to protect these systems from computer threats.

Outlier detection refers to techniques that define and characterize normal or acceptable behaviors of the system (e.g., CPU usage, job execution time, system calls). Behaviors that deviate from the expected normal behavior are considered intrusions [3, 4].

NIDSs can also be divided into two groups depending on where they look for intrusive behavior: *Network-based IDS (NIDS)* and *Host-based IDS*. The former refers to systems that identify intrusions by monitoring traffic through network devices (e.g. Network Interface Card, NIC). A *host-based IDS* monitors file and process activities related to a software environment associated with a specific host. Some *host-based*

IDSs also listen to network traffic to identify attacks against a host [3],[4]. A *false positive* occurs when IDS generates an alarm from normal user activity. If IDS generates too many false positives, then we will lose confidence in the capability of IDS to protect the network. IDS may also experience *false negatives*. In this situation, an attack occurs against the network and IDS fails to alarm even though it is designed to detect such an attack.

2 Dataset Description

Our proposed system is implemented on KDDCUP'99 dataset which was maintained by DARPA, and defined by MIT, USA. Since 1999, KDD'99 [7] has been the most widely used data set for the evaluation of anomaly detection methods. This data set is prepared by Stolfo et al. [5] and is built based on the data captured in DARPA'98 IDS evaluation program [6]. DARPA'98 is about 4 gigabytes of compressed raw (binary) tcpdump data of 7 weeks of network traffic, which can be processed into about 5 million connection records, each with about 100 bytes. The two weeks of test data have around 2 million connection records. KDD training dataset consists of approximately 4,900,000 single connection vectors each of which contains 41 features and is labeled as either normal or an attack, with exactly one specific attack type. We used only 10% of total records. This data set contains four types of intruders which given below:

Denial of Service Attack (DoS): is an attack in which the attacker makes some computing or memory resource too busy or too full to handle legitimate requests, or denies legitimate users access to a machine.

User to Root Attack (U2R): is a class of exploit in which the attacker starts out with access to a normal user account on the system (perhaps gained by sniffing passwords, a dictionary attack, or social engineering) and is able to exploit some vulnerability to gain root access to the system.

Remote to Local Attack (R2L): occurs when an attacker who has the ability to send packets to a machine over a network but who does not have an account on that machine exploits some vulnerability to gain local access as a user of that machine.

Probing Attack: is an attempt to gather information about a network of computers for the apparent purpose of circumventing its security controls.

3 Proposed Framework

The proposed framework is shown in figure 1. This framework is having four layers and the work done towards finding the outliers in each layer is explained below.

Layer 1: Read the training tuples and are described them by n attributes. Each tuple represents a point in n dimensional space. Pre-process the dataset like taking samples and select features subset.

Layer 2: For each selected feature calculate the z-score and set a threshold value to separate the objects into normal and outliers. Here we do this for each attribute.

We take the majority vote to decide whether the object is outlier or normal. For example there are seven attributes and calculated z-scores and set the threshold values. For a record or object four results telling that it is outlier and three results showing it is a normal object, then it is considered as an outlier or intruder because majority votes. At this layer all objects are separated into normal and outliers.

Layer 3: Use the Bayesian network classifier to classify outliers into different classes. In a data set there may be two classes or multiple classes. If there are multiple classes, these are classified using Bayesian Network classifier. In our paper we used kddcup data set which is having five classes. There are four types of attacks, which are described in the dataset description.

Layer 4: Here we read test tuples one after one. We calculate the distance between the test tuple and the previously separated class objects. We determine its type based on its closer ness to the above objects. We used the K-nearest neighbor distance based approach to determine the given object type. All this work is explained below in detail.

3.1 Sampling

Sampling can be used as data reduction technique [8] because it allows a large data set to be represented by a much smaller random sample or subset of data. Suppose that a large dataset, D , contains N tuples. We used simple random sample without replacement (SRSWOR).

Data sets for analysis may contain hundreds of attributes, many of which may be irrelevant to the mining task or redundant. Attributes subset selection reduces the data set size by removing irrelevant or redundant attributes (dimensions). The goal of attribute subset selection is to find a minimum set of attributes such that the resulting probability distribution of the data classes is as close as possible to the original distribution obtained using all attributes. Mining on a reduced set of attributes has an additional benefit. It reduces the number of attributes appearing in the discovered patterns, helping to make the patterns easier to understand. We used Entropy method for features selection. The KDDCUP'99 has 41 features. Using the entropy method 11 features have been selected out of 41 features (attributes/ dimensions) and the selected attributes were listed below.

3.2 Features Selection Using Entropy

This is one of greedy feature selection methods, and conventional information gain [9] which is commonly used in feature selection for classification models. Moreover, our feature selection method sometimes produces more improvements of conventional machine learning algorithms over support vector machines which are known to give the best classification accuracy. It is defined as

$$Entropy (t) = - \sum_j p(j | t) \log_2 p(j | t) \quad (1)$$

The kddcup99 dataset has 41 features. Using the above formulae on the kddcup99 dataset, we selected only top 15 features based on the rank, and the features along their rank were shown in table2, based on entropy of the features which is shown in table1.

Table 1. The features with their rank using Entropy method

Rank	Feature Name	Entropy	Rank	Feature Name	Entropy
1	src_bytes	1.73443	21	dst_host_srv_rerror_rate	0.29025
2	Service	1.60489	22	rerror_rate	0.27679
3	Count	1.53779	23	srv_diff_host_rate	0.26933
4	srv_count	1.17949	24	srv_rerror_rate	0.1916
5	dst_host_same_src_port_rate	1.10496	25	wrong_fragment	0.13855
6	protocol_type	0.98378	26	Hot	0.13141
7	dst_host_diff_srv_rate	0.96876	27	num_compromised	0.08992
8	dst_host_srv_count	0.96519	28	Duration	0.07216
9	dst_host_same_srv_rate	0.89407	29	num_failed_logins	0.02657
10	diff_srv_rate	0.86691	30	Land	0.01799
11	dst_bytes	0.85391	31	root_shell	0.01208
12	same_srv_rate	0.83267	32	is_guest_login	0.00911
13	Flag	0.82963	33	num_file_creations	0.00858
14	dst_host_serror_rate	0.64387	34	num_access_files	0.00734
15	logged_in	0.63707	35	num_root	0.00637
16	serror_rate	0.62279	36	num_outbound_cmds	0
17	dst_host_srv_serror_rate	0.51385	37	is_host_login	0
18	dst_host_count	0.5064	38	Urgent	0
19	srv_serror_rate	0.46685	39	num_shells	0
20	dst_host_srv_diff_host_rate	0.45478	40	su_attempted	0
21	dst_host_rerror_rate	0.35237	41		

Table 2. Selected Features subset using Entropy method

S.No	Selected Feature	S.No	Selected Feature
1	src_bytes	9	dst_host_same_srv_rate
2	Service	10	diff_srv_rate
3	Count	11	dst_bytes
4	srv_count	12	same_srv_rate
5	dst_host_same_src_port_rate	13	Flag
6	protocol_type	14	dst_host_serror_rate
7	dst_host_diff_srv_rate	15	logged_in
8	dst_host_srv_count		

3.3 Z-Score Method

There are many methods for data objects classification. This method is simple and easy to implement for finding outliers. In z-score method, first we calculate the mean and standard deviation values for each attribute, A_i . Using these values zscores can be calculated for each object [10]. It is define as

$$zscore = \frac{v - \bar{A}}{\sigma_A} \quad (2)$$

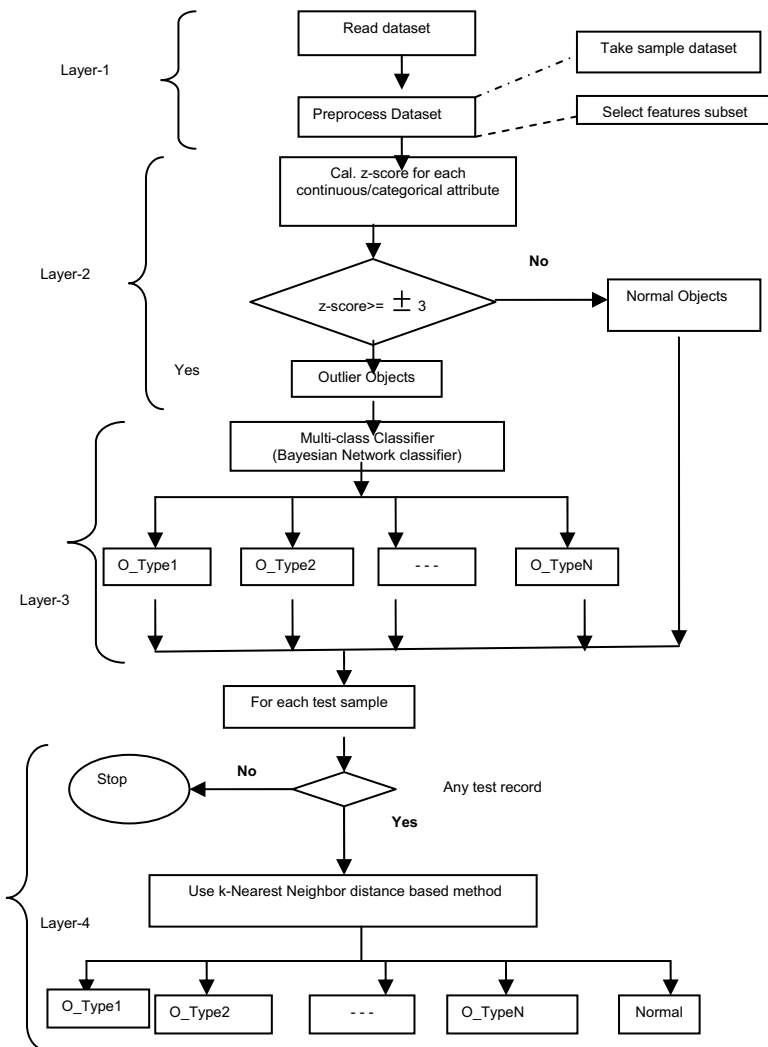


Fig. 1. Proposed Frame work

Table 3. Z-Score values for the attribute src_bytes

protocol	service	flag	src_bytes	Class	z-score
Tcp	http	SF	349	normal.	-0.14754
Tcp	http	SF	349	normal.	-0.14754
Tcp	ftp	SF	350	normal.	-0.14737
Tcp	http	SF	350	normal.	-0.14737
Tcp	http	SF	350	normal.	-0.14737
Tcp	http	SF	350	normal.	-0.14737
Tcp	http	SF	350	normal.	-0.14737
Tcp	http	SF	350	normal.	-0.14737
Tcp	http	SF	350	normal.	-0.14737
Tcp	http	SF	350	normal.	-0.14737
Tcp	http	SF	351	normal.	-0.14721
Tcp	http	SF	352	normal.	-0.14704
Tcp	http	SF	352	normal.	-0.14704
Tcp	http	SF	352	normal.	-0.14704
Tcp	private	REJ	0	ipsweep.	-0.20559
Tcp	smtp	SF	0	ipsweep.	-0.20559
Tcp	private	REJ	0	ipsweep.	-0.20559
Tcp	finger	S0	0	land.	-0.20559
Tcp	finger	S0	0	land.	-0.20559
Tcp	finger	S0	0	land.	-0.20559
Tcp	private	S0	0	neptune.	-0.20559
Tcp	private	S0	0	neptune.	-0.20559
Tcp	private	S0	0	neptune.	-0.20559
Udp	private	SF	28	teardrop.	-0.20093
Udp	private	SF	28	teardrop.	-0.20093
Udp	private	SF	28	teardrop.	-0.20093
Udp	private	SF	28	teardrop.	-0.20093
Icmp	ecr_i	SF	1032	smurf.	-0.03393
Icmp	ecr_i	SF	1032	smurf.	-0.03393
Icmp	ecr_i	SF	1032	smurf.	-0.03393
Icmp	ecr_i	SF	1480	pod.	0.040585
Icmp	ecr_i	SF	1480	pod.	0.040585
Icmp	ecr_i	SF	1480	pod.	0.040585
Icmp	ecr_i	SF	1480	pod.	0.040585
Tcp	ftp_data	SF	7045	warezclient.	0.966234
Tcp	http	RSTR	47612	back.	7.713906
Tcp	http	RSTR	53168	back.	8.638057
Tcp	http	RSTR	53168	back.	8.638057

where v is the value of A . The \overline{A} , and σ_A are the mean and standard deviation, respectively, of an attribute value A . By considering the majority of votes, then that particular object is determined as an outlier based on threshold value. The threshold value is determined as on training samples. The threshold value and z-score value compared to determine outliers. zscore values along the sample dataset for kddcup dataset is shown in table3.

By setting threshold value to compare z-score value, we can separate intruders from normal objects.

3.4 Multiclass Classifier

3.4.1 Bayesian Classifier

One of most effective classifiers in predictive performance is so called naïve Bayesian classifier [11]. This classifier learns from training data the conditional probability of each attribute A_i given the class label C . Classification is then done by applying Bayes rule to compute the probability of C given the particular instance of A_1, \dots, A_n , and then predicting the class with highest posterior probability. This computation is rendered feasible by making a strong independence assumption: all the attributes A_i are conditionally independent given the value of the class C . But this assumption is unrealistic and a better method is Bayesian network.

3.4.2 Bayesian Network Classifier

These networks [12] are directed acyclic graphs that allow efficient and effective representation of the joint probability distribution over a set of random variable. Each vertex represents a random variable, and edges represent direct correlations between the variables. More precisely, the network encodes the following conditional independence statements: each variable is independent of its non-descendants in the graph given the state of its parents. These independencies are then exploited to reduce the number of parameters needed to characterize a probability distribution, and to efficiently compute posterior probabilities given evidence. Probabilistic parameters are encoded in a set of tables, one for each variable, in the form of local conditional distributions of a variable given its parents. Using the independence statements encoded in the network, the joint distribution is uniquely determined by these local conditional distributions.

3.4.3 Learning Bayesian Networks

Consider a finite set $U = \{ X_1, X_2, \dots, X_n \}$ of discrete random variable where each variable X_i may take on values from a finite set, denoted by $\text{Val}(X_i)$. We use capital letters such as X, Y, Z for variable names, and lower-case letters such as x, y, z to denote specific values taken by those variables. Sets of variables are denoted by bold-face capital letters such as $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ and assignments of values to the variables in these sets are denoted by boldface lowercase letters x, y, z . Finally, let P be a joint probability distribution over the variables in U , and let $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ be subsets of U . We say that \mathbf{X} and \mathbf{Y} are conditionally independent given \mathbf{Z} .

A Bayesian network [13] is an annotated directed acyclic graph that encodes a joint probability distribution over a set of random variables U . Formally, a Bayesian

network for U is a pair $B = (G, \Theta)$. The first component, G is a directed acyclic graph whose vertices correspond to the random variables X_1, X_2, \dots, X_n , and whose edges represent direct dependencies between the variables. The graph G encodes independent assumptions: each variable X_i is independent of its nondescendants given its parents in G . The second component of the pair, namely Θ , represents the set of parameters that quantifies the network. A Bayesian network B defines a unique joint probability distribution over U given by [13].

$$P_B(X_1, X_2, \dots, X_n) = \prod_{i=1}^n P_{B_i}(X_i | \Pi_{x_i}) = \prod_{i=1}^n \theta_{x_i} | \Pi_{x_i} \quad (3)$$

3.5 K-Nearest Neighbor (k-NN) Method

There are many distance based method for finding outliers. Knn is one of the distance based approach for classification of objects. The K-Nearest Neighbors (KNNs) model [14] is a very simple, but powerful tool. It has been used in many different applications and particularly in classification or in clustering tasks. K-Nearest Neighbor method is based on learning by analogy, that by computing a given test tuple with training tuples that are similar to it. When given an unknown tuple, a k-NN method searches the pattern space for the k training tuples that are closest to the unknown tuple. Closeness is defined in terms of distance metric, such as Euclidean distance. The Euclidean distance between two points or tuples, say $X_1=(x_{11},x_{12},\dots,x_{1n})$ and $X_2=(x_{21},x_{22},\dots,x_{2n})$, is defined as

$$dist(X_1, X_2) = \sqrt{\sum_{i=1}^n (x_{1i} - x_{2i})^2} \quad (4)$$

We used this distance metric to determine the type of test of tuple. We repeat this for all test samples. The results are shown in the experimental results.

4 Experimental Results

The performance of the proposed model can evaluated using different measures. Some of the measure we explained below.

True positives (TP): refers to the positive tuples that were correctly labeled by the classifier.

True negatives (TN): refers to the negative tuples that were correctly labeled by the classifier.

False positives (FP): are the negative tuples that were incorrectly labeled.

False negatives (FN): are the positive tuples that were incorrectly labeled.

True Positive Rate (TPR) and False Positive Rate (FPR) can be defined in terms of TP, FN, and FP as follows:

$$TPR = \frac{TP}{TP + FN} \quad FPR = \frac{FP}{FP + TN} \quad (5)$$

The other parameter is the *confusion matrix* is more commonly named *contingency table*. In DARPA KDDCUP there are four class types, and therefore we need a 4x4 confusion matrix, the matrix could be arbitrarily large. The number of correctly classified instances is the sum of diagonals in the matrix; all others are incorrectly classified.

The *True Positive (TP) rate (TPR)* is the proportion of examples which were classified as class x , among all examples which truly have class x , i.e. how much part of the class was captured. It is equivalent to *Recall* or *Sensitivity*.

The *False Positive (FP) rate (FPR)* is the proportion of examples which were classified as class x , but belong to a different class, among all examples which are not of class x . In the matrix, this is the column sum of class x minus the diagonal element, divided by the rows sums of all other classes.

The Precision for a class is the number of true positives (i.e. the number of items correctly labeled as belonging to the positive class) divided by the total number of elements labeled as belonging to the positive class (i.e. the sum of true positives and false positives, which are items incorrectly labeled as belonging to the class).

$$Pr = \frac{TP}{TP + FP} \quad (6)$$

F-Measure (F_v) combines the true positive rate (recall) and precision Pr into a single utility function which is defined as $\frac{1}{2}$ -weighted harmonic mean.

Receiver Operating Characteristic (ROC): ROC approach to evaluating predictive ability of classifiers provides an intuitive and convenient way of dealing with asymmetric costs of the two types of errors. ROCs are plotted in coordinates spanned by the rates of false positive and true positive classifications.

Table 4. Performance (measures: TP Rate, FP Rate, Precision , Recall, F_Measure, ROC Area) of the proposed method

Class	TP_Rate	FP Rate	Precision	Recall	F_Measure	ROC_Area
Normal	0.996	0.002	0.996	0.996	0.996	0.997
Dos	0.999	0.002	0.998	0.999	0.999	0.998
Probe	0.969	0.002	0.967	0.969	0.968	0.984
R2l	0.895	0.001	0.885	0.885	0.89	0.947
U2r	0.357	0.0	0.556	0.357	0.435	0.678
Weighted_Avg	0.995	0.002	0.994	0.995	0.995	0.996

Table 5. Confusion Matrix

	Normal	Dos	Probe	R2l	U2r
Normal	3520	3	7	5	0
Dos	3	5975	4	0	0
Probe	6	7	437	0	1
R2l	4	0	2	77	3
U2r	2	0	2	5	5

5 Conclusion

Our proposed framework minimizes the mining task by reducing the dataset size by sampling and reduced feature set which will reduce the processing time and space. Our model is implemented on KDDCUP'99 dataset for evaluating our proposed model. We took 10068 sample records from one million records and we reduced the features from 41 to 15. This will reduce the processing time and space requirement. We got 98% accuracy for the Bayesian Network Classifier and we got 2.7% root mean square error for the K-nearest Neighbor approach. These results showing our model is performing well. This model is not tested on real datasets.

Acknowledgments. We thank the anonymous referees for their careful reading of the paper and their valuable comments that significantly improved its quality.

References

1. Hwang, T.S., Lee, T.-J., Lee, Y.-J.: A Three-tier IDS via Data Mining Approach. In: MineNet 2007, San Diego, California, USA, June 12 (2007)
2. Srinivasulu, P., Nagaraju, D., Ramesh Kumar, P., Nageswara Rao, K.: Classifying the Network Intrusion Attacks using Data Mining Classification Methods and their Performance Comparison. IJCSNS International Journal of Computer Science and Network Security 9(6) (June 2009)
3. McHugh, J.: Intrusion and Intrusion Detection. Technical Report CERT Coordination Center, Software Engineering Institute, Carnegie Mellon University (2001)
4. Bezroukov, N.: Intrusion Detection (general issues). Open Source Software Educational Society, Softpanorama (July 19, 2003), http://www.softpanorama.org/Security/intrusion_detection.shtml (October 30, 2003)
5. Stolfo, S.J., Fan, W., Lee, W., Prodromidis, A., Chan, P.K.: Costbased modeling for fraud and intrusion detection: Results from the jam project. Discex 2, 1130 (2000)
6. Lippmann, R.P., Fried, D.J., Graf, I., Haines, J.W., Kendall, K.R., McClung, D., Weber, D., Webster, S.E., Wyszogrod, D., Cunningham, R.K., Zissman, M.A.: Evaluating intrusion detection systems: The 1998 darpa off-line intrusion detection evaluation. Discex 2, 1012 (2000)

7. KDD Cup 1999. (October 2007),
<http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>
8. Bronnimann, H., Chen, B., Dash, M., Haas, P., Qiao, Y., Scheuermann, P.: Efficient Data-Reduction Methods for On-Line Association Rule Discovery in thesis
9. Lee, C., Lee, G.G.: Information gain and divergence-based feature selection for machine learning-based text categorization. In: Information Processing & Management, vol. 42(1), pp. 155–165 (January 2006)
10. <http://www.measuringusability.com/zcalc.htm>
11. Langley, P., Iba, W., Thompson, K.: An analysis of Bayesian classifiers. In: Proceedings of the Tenth National Conference on Artificial Intelligence, pp. 223–228. AAAI Press, San Jose (1992)
12. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian Network Classifiers, vol. 29(2-3), pp. 131–163. ACM, New York (November/December 1997)
13. Susanne, G., Dethlefsen, B.C.: Learning Bayesian Networks with R. In: Proceedings of the 3rd International Workshop on Distributed Statistical Computing (DSC 2003), Vienna, Austria, March 20–22 (2003)
14. Song, Y., Huang, J., Zhou, D., Zha, H., Lee Giles, C.: IKNN: Informative K-Nearest Neighbor Pattern Classification. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenić, D., Skowron, A. (eds.) PKDD 2007. LNCS (LNAI), vol. 4702, pp. 248–264. Springer, Heidelberg (2007)

A New Routing Protocol for Mobile Ad Hoc Networks

S. Rajeswari¹ and Y. Venkataramani²

¹ Associate professor, Dept. of Electronics and Communication Engineering,
Saranathan college of Engineering

rajee_ravi@sify.com

² Principal, Saranathan college of Engineering

diracads@saranathan.ac.in

Abstract. Mobile Adhoc Networks (MANETs) is a wireless infrastructure-less network, where nodes are free to move independently in any direction. The nodes have limited battery power; hence we require energy efficient routing protocols to optimize network performance. This paper aims to develop a new routing algorithm based on the energy status of the node. In this paper, it has been proposed a new Protocol for Mobile Adhoc Networks which is expected to achieve energy efficiency and reliability. The gossiping probability of a node is determined by the throughput. Differently from other proposals no external location service support, e.g., via GPS, is not required. Rather, the throughput is estimated from the “inside” of the network using feedback factor through the node propagation to the destination and the value of gossiping probability is adaptively adjusted. This results in less energy consumption and more reliability in the network.

Keywords: Ad hoc Networks, Throughput, Delivery Ratio, Energy Consumption, Reliability.

1 Introduction

A mobile ad-hoc network (MANET) is a collection of many mobile nodes with no infrastructure. To form a network over radio links, the mobile nodes are self-organized. Extending mobility into the self-organized, mobile and wireless domains is the main objective of MANETs where a set of nodes form the network routing infrastructure in an ad-hoc fashion. MANETs are used in those areas where wired network is unavailable and where rapid deployment and dynamic reconfiguration are necessary. These include military battlefields, emergency search, rescue sites, classrooms and conventions, where the participants share information dynamically using their mobile devices [1]. Generally the existing routing protocols are classified as Table-driven (proactive) routing protocol, On-demand (reactive) routing protocol, Hybrid routing protocol and Geographic routing protocol.

1.1 Proactive MANET Protocol (PMP)

A proactive MANET protocol (PMP) detects the layout of the network which is active. PMP maintains a routing table at every node. From the routing table, with minimal

delay, a route can be determined. The PMP can provide good reliability and low latency. This protocol cannot update the route information immediately for a node moving with high speed. Also for a node moving occasionally, updating the unchanged entry continuously in the routing table results in much traffic overhead and wastage of network resources. PMP is not appropriate for large scale MANETs [2]. PMP is used in DSDV [11], OLSR [14].

1.2 Reactive MANET Protocol (RMP)

In Reactive MANET protocol (RMP) when the source node requests communicate with the other node, only then a route between a pair of nodes is found. For nodes with high mobility and for nodes which transmit data occasionally, this on-demand approach is quite suitable. But in RMP, the source node broadcasts route requests throughout the network and has to wait for the response which is a disadvantage. This route discovery procedure results in a major delay [2]. RMP is used in, DSR[12], AODV [13] and TORA[15].

1.3 Hybrid Routing Protocol

It is distinct that both PMP and RMP have their own merits and drawbacks [2]. It is hence natural to consider a hybrid approach that integrates the merits of PMP and RMP. Zone routing protocol (ZRP) [2] common characteristic is that the association of nodes with zones is generally deterministic, i.e., each node must be in a specific zone. The intra-zone routing protocol is proactive while the inter-zone routing is performed by the reactive approach.

1.4 Geographic Routing Protocol

Geographic routing uses nodes' locations as their addresses, and forwards packets (when possible) in a greedy manner towards the destination. One of the key challenges in geographic routing is how to deal with dead-ends, where greedy routing fails because a node has no neighbor closer to the destination (*GPSR*) [18].

1.5 Gossip Routing Protocol

For location discovery or for secure routing applications, most ad hoc routing algorithms depend on broadcast flooding. Though flooding is a robust algorithm, because of its extreme redundancy, it is unfeasible in dense networks. The use of flooding algorithms may lead to broadcast storms in large wireless networks where the number of collisions is so high it could cause system failure. Since the packet retransmission is based on the outcome of coin tosses, Gossip [3] is a probabilistic algorithm. The main objective of gossip is to minimize the number of retransmissions, while maintaining the main benefits of flooding.

The paper is organized as follows. In Section 2 we summarize related works. Section 3 presents an analytical performance model of the gossip protocol. Section 4 describes an implementation of the algorithm which exploits throughput as feedback; simulation results are provided in Section 5 and concluding remarks in section 6.

2 Related Work

Xiaobing Hou and David Tipper [3] propose the Gossip-based Sleep Protocol (GSP). With GSP, each node randomly goes to sleep for some time with gossip sleep probability p . When the value of p is small enough, the network stays connected. Mike Burmester et al. [4] consider ways to reduce the number of redundant transmissions in flooding while guaranteeing security. They present several new gossip protocols which automatically correct all faults and guarantee delivery. Yuval Shavitt and Amir Shay [5] introduce the Gossip Network model where travelers can obtain information about the state of dynamic networks by gossiping with peer travelers using ad-hoc communication. Travelers then use the gossip information to re-choose their path and find the shortest path to destination. Zhongmin Shi and Hong Shen [6] Gossip-based techniques recently adopted in mobile ad hoc network (MANET) system have achieved significant improvement on network overhead, routing efficiency and reliability. Zygmunt J. et al. [7], propose a gossiping-based approach, where each node forwards a message with some probability, to reduce the overhead of the routing protocols. With less number of executions maximum of nodes get the message that depends on the gossiping probability and the topology of the network. Hany Morcos et al. [8] propose a new approach - by control-theoretic adaptations similar to those widely used in the Internet, e.g. additive-increase multiplicative-decrease (AIMD) of TCP for reacting to congestion conditions. Cigdem Sengul Indranil Gupta et al. [9] propose networking protocols for multi-hop wireless sensor networks (WSNs) which are required to simultaneously minimize resource usage as well as optimize performance metrics such as latency and reliability. They explore the energy-latency-reliability trade-off for broadcast in multi-hop WSNs, by presenting a new protocol called PBBF (Probability-Based Broadcast Forwarding). Xiang-yang li et al.[10] used Gossip based routing method and re-checked to reduce the number of messages in both wired networks and wireless ad hoc networks.

3 Gossip Routing in Adhoc Networks

A message is normally transmitted as a broadcast rather than a unicast communication in adhoc networks. So the message is received by the entire nodes one hop away from the sender. Since wireless resources are expensive, we use this physical-layer broadcasting feature of the radio transmission. In the gossiping protocol, we control the probability with which this physical-layer broadcast is sent [7]. The basic gossiping protocol is simple. A source sends a route request with probability 1. When a node first receives a route request, with probability p it broadcasts the request to its neighbors and with probability $1-p$ it discards the request; if the node receives the same route request again, it is discarded. Thus, a node broadcasts a given route request at most once [6]. Thus, in almost all executions of the algorithm, either scarcely any nodes receive the message, or most of them do. Ideally, we could make less number of executions where the gossip dies out relatively low while also keeping the gossip probability low, to reduce the message overhead [7].

The gossip routing protocol satisfies the following conditions:

- The main portion of the protocol involves periodic, pairwise, inter-process communications.
- During these communications the information exchanged is of bounded size. Agents interact, just to intimate the change in the state of the other agent.
- A gossip communication does not occur when A pings B, to compute the response time, as this does not involve the transmission between agents.
- Reliable communication is not implicit.
- The protocol costs are insignificant since the frequency of the communications is low compared to classic message latencies.

Since the current ad hoc network routing protocols require all the nodes to be awake and keep listening wastes a lot of energy.

4 A New Proposed Routing Protocol

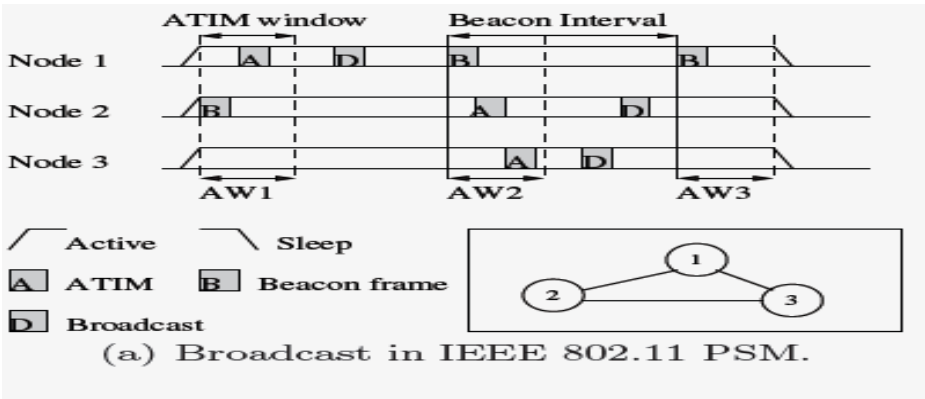
Our observation is that if gossiping can make all the nodes receive a message, then the nodes forwarding the message are connected at least by the paths the message passes through. Therefore, in a static network without mobility (e.g., a sensor network), with certain probability p' , gossiping protocols [7] can make almost all nodes in the network receive the message. Then if all nodes go to sleep with probability $p = (1-p')$, almost all the awake nodes stay connected. Thus, we can safely put a percentage (p) of the nodes in sleep mode without losing network connectivity. We term p the gossip sleep probability.

Let us assume that every node in the ad hoc network chooses an equally distributed random time interval, known as gossip interval. When the time is up, the node will immediately choose another random interval independently. To make it possible, we assume the feasible maximum gossip interval is much smaller than the lifetime of the network. Each node independently generates a random time interval and chooses either going to sleep with probability p or staying awake with probability $1-p$ for the interval. Every sleeping node wakes up at the end of its interval. Every node repeats the above process for every random interval independently.

4.1 Sleep Scheduling Mechanisms

There are two main MAC-layer approaches to reduce energy consumption in Adhoc Networks. The first approach is to use an active-sleep cycle, which lets nodes sleep periodically. The second approach involves using an additional low-power wake-up radio to wake up nodes. However, since this approach requires an extra hardware component on the sensor node, the remainder of the paper focuses on only the active-sleep cycle approach.

The basic idea of introducing an active-sleep cycle to a contention-based protocol is to divide time into frames. Each frame is divided into an active time and a sleep time. During the sleep time, a node puts its radio in sleep mode to save energy. During the active time, a node can send and receive messages. For instance, the IEEE 802.11 protocol [IEEE 802.11 1999] provides such a power-save mode (PSM), which requires nodes to be time-synchronized and follow the same active sleep schedule. In IEEE 802.11 PSM active and sleep times are fixed.



Since the focus of this paper is broadcast, we next discuss the behavior of these sleep scheduling mechanisms for this communication pattern. Fig.1 shows where nodes are synchronized to wake up at the beginning of every beacon interval. Pending traffic is announced via ATIMs (Adhoc Traffic Indication Messages) in an ATIM window. In the example, Node 1 announces a broadcast ATIM for which all one-hop nodes (or neighbors) (e.g., Node 2 and Node 3) should stay awake to receive the message after the ATIM window.

An immediate observation is that to rebroadcast the message, a node must wait for the next ATIM window to guarantee that each neighbor receives the ATIM advertising the broadcast. This increases latency. A second observation is that when, say, Node 2 retransmits the broadcast message, Node 1 and Node 3 receives redundant packets. Furthermore, due to redundant broadcast packets, nodes stay awake the entire beacon interval more often, mostly listening on the channel. This increases energy consumption.

Hence, broadcast traffic does not increase the energy spent in idling; however, energy consumption still increases due to redundancy. Additionally, nodes that follow more than one schedule add to redundancy since these nodes typically transmit a broadcast message multiple times to guarantee the neighbors with different schedules receive the message. However the broadcast redundancy problem remains as all nodes send the message once to IEEE 802.11 PSM. For each broadcast packet, nodes need to send and receive a preamble at least as long as the check interval. Hence, the energy spent for listening increases with redundant packets. Therefore, the sleep scheduling mechanisms for Adhoc networks display similar disadvantages in the presence of broadcast traffic. Motivated by these observations, we propose a new Routing Gossip Routing Protocol which allows achieving less energy consumption and more reliability.

A node (which wants to communicate) maintains a control variable called C which represents the active number of neighbors. The rest of the nodes in the network will be in either p or 1-p state. The higher - C is the more power the node uses to send packets and thus the communication is more reliable. When node X needs to broadcast a data packet, X looks up its neighbor list for the distance between itself and its neighbors numbered C. X then calculates the amount of power needed to send the

packet to that neighbor. Every node initializes C to one. This means that a node initially broadcasts data packets only to its closest neighbor, thus requiring the least power. After sending data packet, node X waits for a feedback from destination. While receiving packets at the destination, the throughput D is calculated and it will be sent as a feedback to the source. If X hears a feedback D for the data packet below a reliability threshold RT , X increases the value of C there by increasing the probability of number of active nodes. This assures the increased throughput. When D becomes greater than or equal to RT , the value of C is decreased adaptively to decrease the number of forwarding nodes and there by decrease the probability of number of active nodes. This process continues until either X hears a feedback for the packet or the value of C reaches reliability threshold RT , which is determined by the total number of neighbors. Upon receiving a feedback, X starts to decrease the value of C (after a certain number of acknowledged data packets) to a minimum value of one.

Algorithm

1. Let sleep probability $P(s) = p$ and awake probability $A(s) = 1-p$.
2. Let initial value of $C=1$.
3. X broadcasts data packets to Y .
4. At Y , calculate throughput D ,
 $D = \text{Number of packets received} / \text{Number of packets sent}$
5. Y sends D as a feedback to X .
6. At X , If $D < RT$ then,
 - 6.1. $C = C + \delta$, where δ is the additive factor.
 - 6.2. Repeat from 3.
7. End
8. Else. If $D \geq RT$, then,
 - 7.1. If $C > 1$, then,
 - 7.1.1. $C = C - \delta$
 - 7.1.2. Repeat from 3.
 - 7.2. End If
9. End If
10. End.

After some beacon interval of time, from the step 3, algorithm will be repeated.

To summarize, new routing protocol has the following salient features:

- Unlike existing routing schemes, new protocol is neither single-path nor multi-path; rather each node exploits the multiplicity of paths based on its observed loss conditions.
- In new protocol, only for low packet delivery ratios, a node uses high-power transmissions to reach farther neighbors. For high packet delivery ratios, a node adapts to low-power transmissions. Thus, new protocol sensibly consumes power based on local error conditions, which maximizes the lifetime of the network and minimizes the cost of the power consumed per successfully delivered data.
- Thus new protocol aggressively probes for possible routes to deliver data packets, thus reacting quickly within unreliable areas of the network.

5 Simulation Results

5.1 Simulation Model and Parameters

NS2 is used to simulate the proposed algorithm. In our simulation, the channel capacity of mobile hosts is set to the value: 2 Mbps. For the MAC layer protocol the distributed coordination function (DCF) of IEEE 802.11 is used. It has the functionality to notify the network layer about link breakage. In the simulation, mobile nodes move in a 800 meter x 400 meter region for 50 seconds simulation time. The number of mobile nodes is kept as 40. We assume each node moves independently with the same average speed. All nodes have the same transmission range of 250 meters. In our simulation, the speed is set as 10m/s. The simulated traffic is Constant Bit Rate (CBR). The pause time of the mobile node is kept as 10 sec.

5.2 Performance Metrics

Our proposed a new Routing protocol is compared with existing Probability Based Broadcast Forwarding GSP [9] protocol. The evaluation is mainly based on performance according to the following metrics:

Throughput: It is the number of packets received successfully.

Average Energy: It is the average amount of energy consumption of all nodes in sending, receiving and forward operations expressed in Joules.

Drop: It is the number of packets dropped.

Packet Delivery Fraction: It is the ratio of the fraction of packets received successfully to the total number of packets sent.

5.3 Results

1. Based on number of neighbor nodes

In our initial simulation, we vary the number of flows as 1, 2, 3 and 4.

As we can see from the figure, the throughput is more in the case of new protocol than existing protocol.

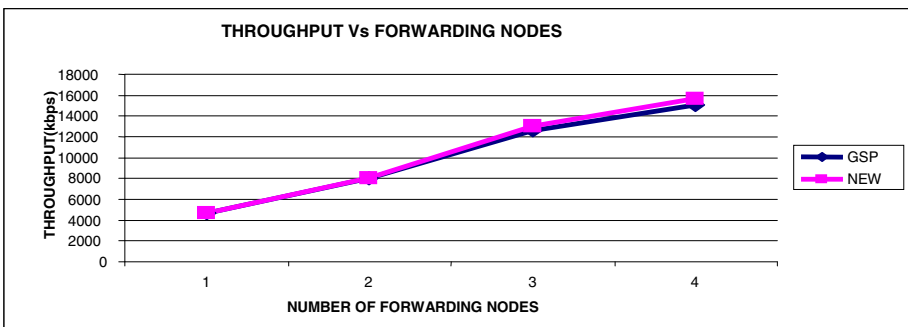


Fig. 1. Throughput of protocols when the number of flow is increased

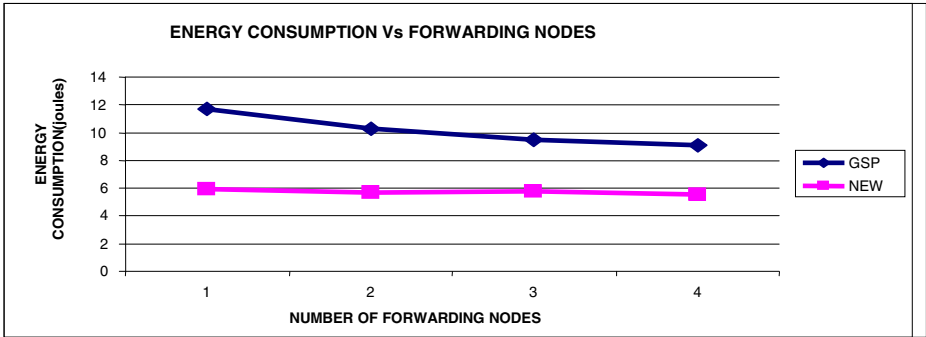


Fig. 2. Energy consumption for varying number of nodes

From the results, we can see that propose scheme has less energy than existing scheme, since it has the energy efficient routing.

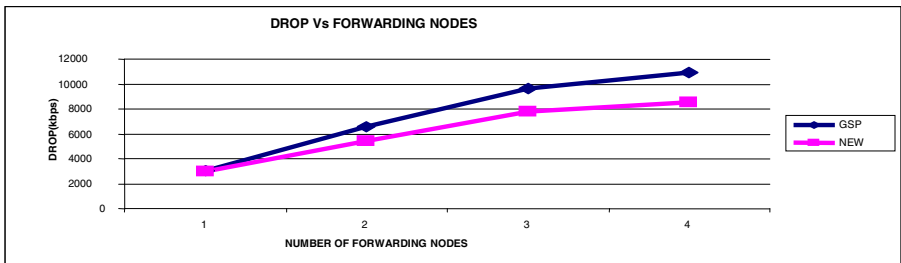


Fig. 3. Packets drop with forwarding number number of nodes

From the figure we can show that the packets dropped are less for proposed protocol when compared with existing protocol.

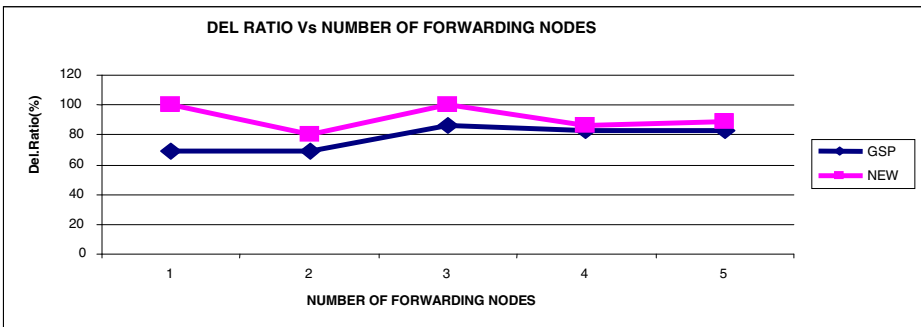


Fig. 4. Packet delivery ratio with forwarding nodes

This Figure gives the comparison for both the protocols. Since the packet drop is less and the throughput is more, proposed scheme achieves good delivery ratio, compared to existing scheme.

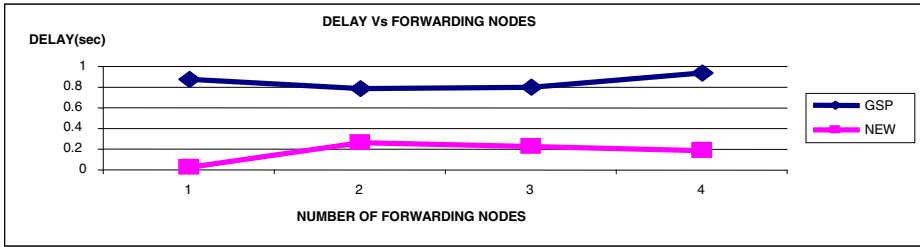


Fig. 5. Delay with forwarding nodes

From the figure, the number of packet dropped are less in new protocol compared to existing protocol.

2. Based on Probability Value

In the second simulation experiment, we vary the probability values as 0.0, 0.1, 0.2, 0.3 and 0.4.

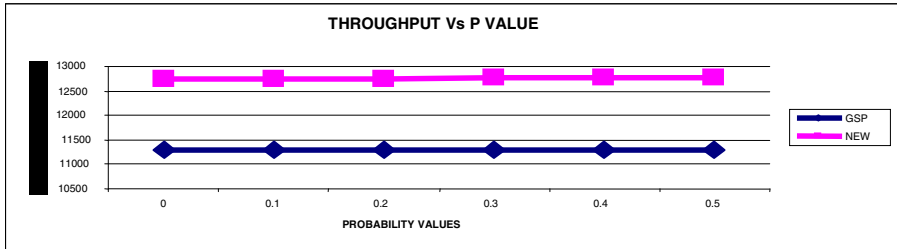


Fig. 6. Throughput for varying number of active nodes

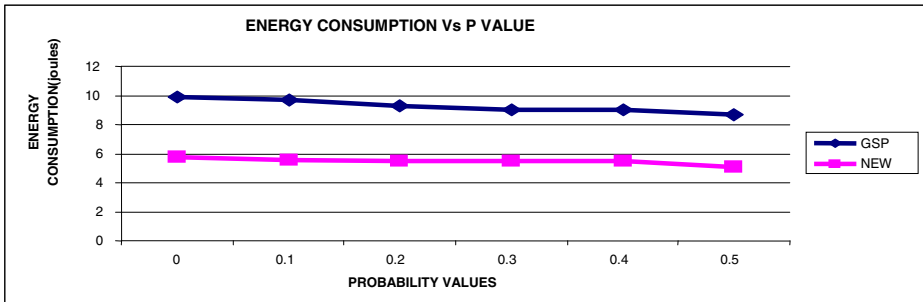


Fig. 7. Energy consumption (Joules) for varying probability values

Figure gives the throughput of both the protocols when the number of Probability values is increased. As we can see from the figure, the throughput is more in the case of proposed scheme than existing scheme.

Figure shows the results of energy consumption for the number of p values as 0.0, 0.1, 0.2, 0.3 and 0.4. From the results, we can see that propose scheme has less energy consumption than existing scheme, since it has the energy efficient routing.

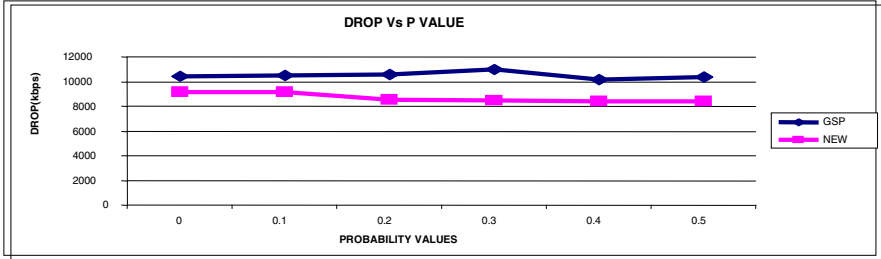


Fig. 8. Number of packets dropped with increasing probability values

We can ensure that the packets dropped are less for proposed protocol when compared with existing protocol.

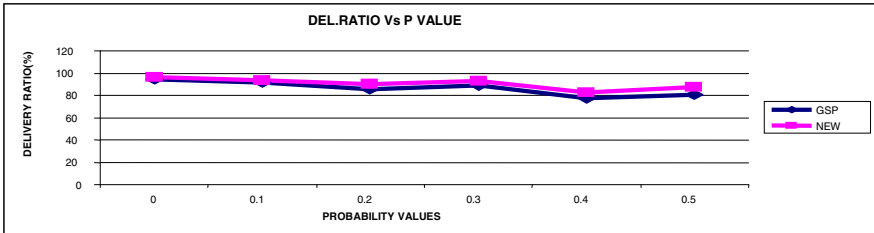


Fig. 9. Delivery Ratio for varying number of active nodes

This figure presents the packet delivery ratio of both the protocols. Since the drop is less and the throughput is more, proposed scheme achieves better delivery ratio, compared to existing protocol.

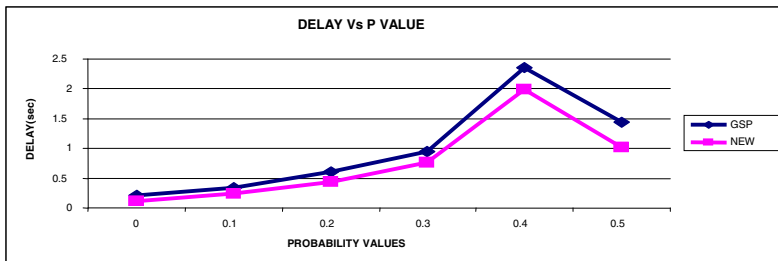


Fig. 10. Delay for varying probability values

Figure shows the results of dropped packets for varying number of probability values as 0.0, 0.1, 0.2, 0.3 and 0.4. From the results, we can see that proposed scheme has less delay than existing scheme, since it has the efficient routing.

6 Based on Rate

In the third simulation experiment, we vary the data sending rate as 100, 200, 300, 400 and 500 kbps.

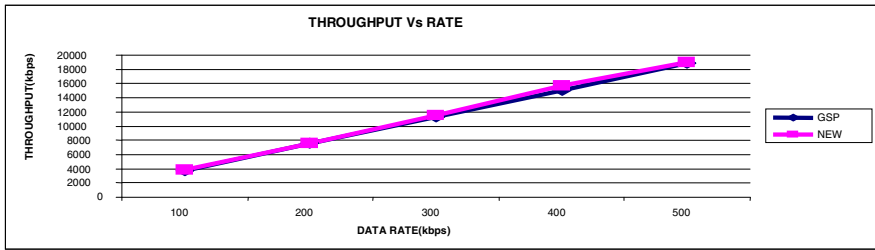


Fig. 11. Throughput for varying data rates

Figure gives the throughput of both the protocols when the number of flow is increased. As we can see from the figure, the throughput is more in the case of proposed protocol than existing protocol.

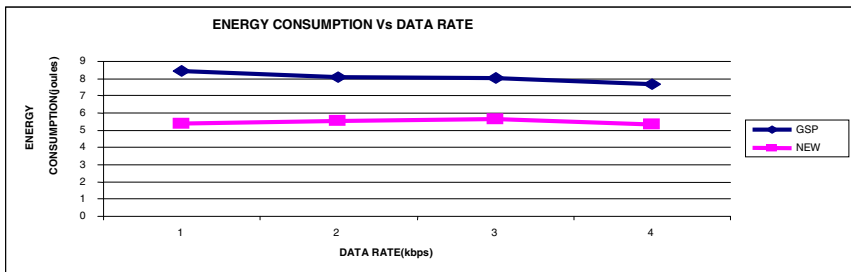


Fig. 12. Energy consumption for varying data rates

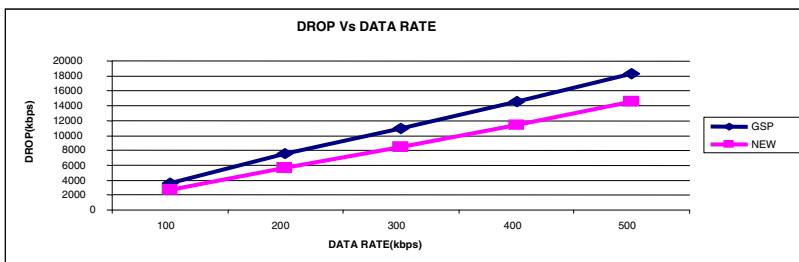


Fig. 13. Packets dropped for varying data rates in kbps

Figure shows the results of energy consumption for different varying data rates. From the results, we can see that proposed scheme has less energy consumption than existing scheme, since it has the energy efficient routing.

We can ensure that the packets dropped are less for proposed scheme when compared with existing scheme.

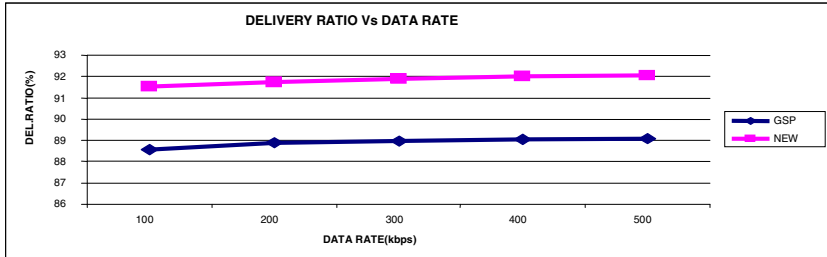


Fig. 14. Packets delivery Ratio for varying data rates

Figure presents the packet delivery ratio of both the protocols. Since the packet drop is less and the throughput is more, proposed could able to achieve good delivery ratio, compared to existing protocol.

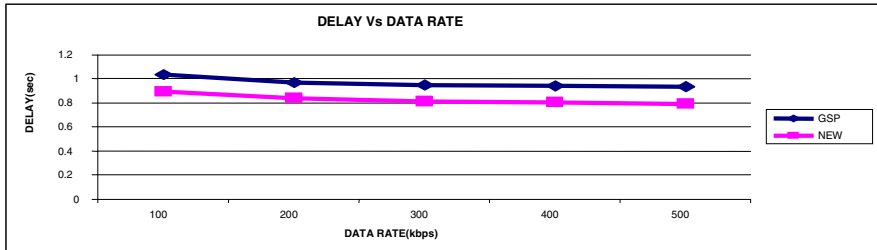


Fig. 15. Packets delay for varying data rates

7 Conclusion

This paper introduces a new class of gossip algorithms designed to direct the gossip process towards a specific polarizing node in the network. The algorithm has been adaptive in Gossip. While in the classical gossip algorithm each node forwards a message with the same probability, our proposal is characterized by a variable gossiping probability-based on the throughput, which is high enough only for sustaining the spreading process towards the polarizing node. An important application of the algorithm is when used for path discovery in MANET. We have shown by simulations the protocol allows saving up to 80% of energy efficiency compared to a pure flooding, while 60% of nodes have to process a requesting packet.

References

1. Junhai, L., Liu, X., Danxia, Y.: Research on multicast routing protocols for mobile ad-hoc networks. Elsevier, Science Direct, Computer Networks, pp. 988–997 (2007)
2. Xie, J., Quesada, L.G., Jiang, Y.: A Threshold-based Hybrid Routing Protocol for MANET. In: IEEE ISWCS (2007)
3. Hou, X., Tipper, D.: Gossip-based Sleep Protocol (GSP) for Energy Efficient Routing in Wireless Ad Hoc Networks. In: IEEE Xplore, Wireless Communications and Networking Conference, March 21–25, vol. 3, pp. 1305–1310 (2004)
4. Burmester, M., Van Le, T., Yasinsac, A.: Adaptive gossip protocols: managing security and redundancy in dense ad hoc networks. In: Ad Hoc Networks, April 2007, vol. 5(3), pp. 313–323. Elsevier Science Publishers, Amsterdam (2007)
5. Shavitt, Y., Shay, A.: Optimal Routing in Gossip Networks. IEEE Transactions on Vehicular Technology 54(4), 1473–1487 (2005)
6. Shi, Z., Shen, H.: Adaptive Gossip-based Routing Algorithm. In: IEEE International Conference on Performance, Computing, and Communications, pp. 323–324 (2004)
7. Haas, Z.J., Halpern, J.Y., Li, E.L.: Gossip-Based Ad Hoc Routing. IEEE ACM Transactions on Networking 14(3), 479–491 (2006)
8. Morcos, H., Matta, I., Bestavros, A.: M2RC: Multiplicative-increase/additive-decrease Multipath Routing Control for Wireless Sensor Networks, ACM, Technical Report BUCS-TR-2004-029, Computer Science Department, Universidad de Boston (2004)
9. Miller, M.J., Sengul, C., Gupta, I.: Exploring the Energy-Latency Trade-off for Broadcasts in Energy-Saving Sensor Networks. In: 25th IEEE International Conference on Distributed Computing Systems (ICDCS 2005), June 6–10, pp. 17–26 (2005)
10. li, X.-Y., Moaveninejad, K., Frieder, O.: Regional Gossip Routing for Wireless Ad Hoc Networks. In: Mobile Networks and Applications, vol. 10, pp. 61–77. Kluwer Academic Publishers, Dordrecht (2005)
11. Perkins, C.E., Bhagwat, P.: Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In: ACM SIGCOMM Computer Communication Review, Association of Computing Machinery Publishers, vol. 24(4), pp. 234–244 (1994)
12. Johnson, D.B., Maltz, D.A.: Dynamic Source Routing in Ad Hoc Wireless Networks. In: Mobile Computing. ch. 5, pp. 153–181. Kluwer Academic Publishers, Dordrecht (1996)
13. Perkins, C.E., Bhagwat, P.: Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers. In: ACM SIGCOMM Computer Communication Review, vol. 24(4), pp. 234–244 (October 1994)
14. Perkins, C.E., Royer, E.M.: Ad-hoc On-Demand Distance Vector Routing. In: WMCSA Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications, pp. 90–100. IEEE Computer Society Publishers, Los Alamitos (February 1999)
15. Clausen, T., Jacquet, P.: Optimized Link State Routing Protocol (OLSR). In: Network Working Group, Proceedings of the IEEE INMIC, Pakistan (October 2001)
16. Park, V.D., Scott Corson, M.: A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks. In: INFOCOM (Proceedings of the INFOCOM 1997). Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies, Driving the Information Revolution, p. 1405. IEEE Computer Society Publishers, Los Alamitos (1997)
17. Beraldi, R.: The polarized gossip protocol for path discovery in MANETs. Ad Hoc Networks 6(1), 79–91 (2008)
18. Huang, Y., Handurukande, S., Bhatti, S.: Autonomic MANET Routing Protocols. Journal of Networks 4(8), 743–753 (2009)
19. Rao, A., Ratnasamy, S., Papadimitriou, C., Shenker, S., Stoica, I.: Geographic routing without location information. In: MobiCom 2003, San Diego, California, USA, September 14–19 (2003)

Remote-Memory Based Network Swap for Performance Improvement

Nirbhay Chandorkar¹, Rajesh Kalmady¹, Phool Chand¹, Anup K. Bhattacharjee²,
and B.S. Jagadeesh¹

¹ Computer Division, Bhabha Atomic Research Centre, Trombay, Mumbai

² Reactor Control Division, Bhabha Atomic Research Centre, Trombay, Mumbai
{nirbhayc, phool, rajesh, anup, jag}@barc.gov.in

Abstract. On High Performance Computing Clusters we run scientific application programs which are highly memory intensive. Such applications require large amount of primary memory, the execution of which causes swapping on to the secondary memory of the cluster nodes. As the swapped blocks reside on the secondary memory (disk), any access to these affects the throughput of the applications significantly. In this paper we discuss the design and development of a fast and efficient swap device. This device is a network based, pseudo block device, developed to improve the performance of memory intensive work loads by dynamically unifying free physical memory of various nodes of a cluster over the network. We also discuss the performance observed over different network interconnects.

Keywords: Block Device, Remote Memory, Swap Device, Cluster.

1 Introduction

With the increase in availability of greater processing power, scientists are attempting to develop and solve high-resolution models to make accurate simulations of problems that they are trying to investigate. Applications like these need, in addition to huge processing power, large primary memory to give results in a reasonable time frame. But, the primary memory on a commodity computer is always limited owing to the cost trade-offs governed by the number of memory slots and the density of the memory-modules that go into them.

When the over-all memory requirement of all the active processes in a system exceeds the primary memory present on the system, swapping is employed to free some of the memory blocks (as decided by a pre-determined replacement policy) on to the swap space on a secondary storage (disk) to make them available for an active process that has an immediate requirement of memory. Swap space is a space on the disk, which is used by the operating system as an extension to the primary memory. Because of the large disparity in the access times of blocks in primary memory and the blocks that are to be brought from the swap space on the secondary storage, the performance of the program gets reduced drastically when swapping is employed by the operating system. This assumes even more significance as most of the High Performance Computing systems are built by interconnecting commodity processors having

limited memories through commodity interconnects and execute multiple jobs at a given time. Even though the memory available on each of the processors is limited, the aggregation of the primary memory of all the processors in a cluster is indeed substantial.

It is worth noting that current-day clusters typically have at least 256 nodes with each node having about 16-32 GB of Memory and are interconnected using high speed, low latency networks. With such large number of nodes present in a cluster and the memory requirement of programs being executed in them being dynamic in nature, free, unused memory will be available on some of the cluster nodes. We have explored the idea of utilizing the free-memory of the nodes of a cluster for storing the swapped out pages of the other nodes.

2 Related Works

Availability of free memory on work stations have been explored in [1, 2]. The concept of utilizing remote memory started for network of workstations with the high bandwidth, low-latency networks becoming affordable. Several studies [3, 4, 5, 6] have been made to utilize the remote memory over the network. Feeley et. al. [3] have implemented a system in which remote memory is used as the cache for swapped pages that are also written through to disk. They modified the memory management system of the DEC OSF/1 kernel. Their servers only cache clean pages and arbitrarily drop a page when memory resources become scarce. Markatos and Dramitinos [4] proposed the reliability schemes for a remote memory pager for the DEC OSF/1 operating system. Their system is implemented as a client block device driver, and a user-level server for storing pages from remote nodes. For providing reliability they have considered mirroring and parity logging based schemes and further the requests can also be forwarded to disk or disk based file. Bernard and Hamma's work [5] focuses on policies for balancing a node's resource usage between remote paging servers and local processes in a network of workstations. Anderson and Neefe [6] describe the common assumptions about the network RAM and have discussed the possible ways of its implementation. They have further discussed their user level implementation which required modification in code of memory allocator to allocate remote memory.

With further increase in disparity between network and disk access times with introduction of network technologies based on InfiniBand or 10G Ethernet, the concept of utilization of remote physical memory is being increasingly investigated [7, 8, 9].

3 System Design

In order to utilize primary memory of remote machines, we had to develop two basic components in our design. The first component, which runs on the node where the user application gets executed and requires remote memory, is the client part of the system and the other component which runs on the remote nodes serving their memory to the application is the server part of the system.

3.1 Design Issues

To design the system, we have considered the following design issues:

a) Kernel level vs. User level design. The implementation to utilize remote memory can be realized at different levels such as in user space or kernel space. In the user space design the client will have to act as an improvised dynamic memory allocator that allocates memory from remote nodes. In this design scheme we cannot achieve transparency in user level applications, because the applications will have to be written to access the functions used in the implementation. The user level design incurs more software overheads, because of the memory protection mechanism, in which user level programs cannot access the kernel memory directly; instead the data is copied from user to kernel space before transfer. Moreover, if the server were to be in user space, the memory allocated by it might as well get swapped on to the nodes' secondary memory. Hence, we chose to implement the system entirely in kernel space to achieve complete transparency to user applications.

b) Memory allocation scheme. The server module needs to allocate free memory on its node for clients on other nodes. Different policies can be adopted for this. One way is to make the server to allocate available memory pages whenever a swap-out request comes from a client and then free it when it is required no more. But there are several causes for concern present here such as, the additional memory allocation overhead for each request, the possibility of allocation failure and the like. Moreover, the Linux kernel never notifies the swap device that a particular page is no longer needed and can be freed. The other option is to make the server module reserve some fixed amount of memory for the client, when it asks for the memory to initialize the device. The drawback of this approach is that this reserved memory cannot be utilized by other programs on the server node. We chose to implement the second policy, because it does not have risk of memory allocation failure as weighed against the cost of free memory being blocked on the server side.

The block diagram of the system is depicted in Figure 1. The client module is implemented as a network based pseudo block device driver. The server has been developed as a loadable kernel module. The server allocates memory in multiples of fixed size memory chunk, called a "*Swap Block*". Hence, the size of the device can be in multiples of the *Swap Block* size only. At a given time a cluster node either acts as a server or a client. Here onwards the proposed system consisting of two modules, client and server will be referred as Network RAM Block Device (NRBD).

3.2 Client Module

The client module is implemented as the block device driver and Linux kernel's swapping mechanism communicates with this module exactly in the same manner as it would with any hard disk driver. Client module consists of two layers, layer one accepts swap-in swap-out request from the kernel Block I/O layer and forwards the request to second layer, which consists of multiple kernel threads to process the requests. The second layer also has a Main Thread which runs as the daemon; it initializes the pseudo device and acts as an interface between user space and Client Module.

Client's Main Thread initializes the pseudo block device, it negotiates with the servers in the cluster nodes and gets allocated the amount of memory required to create the device. Client keeps track of allocated *Swap Blocks* distributed over the network using its *Swap Map* and *Server List*. *Swap Map* maps contiguous pseudo block device to the distributed *Swap Block* allocated on different nodes. *Server List* has one entry for each Server node which has allocated memory for the Client. Its each node has following fields: request queue, worker thread, pointer to socket.

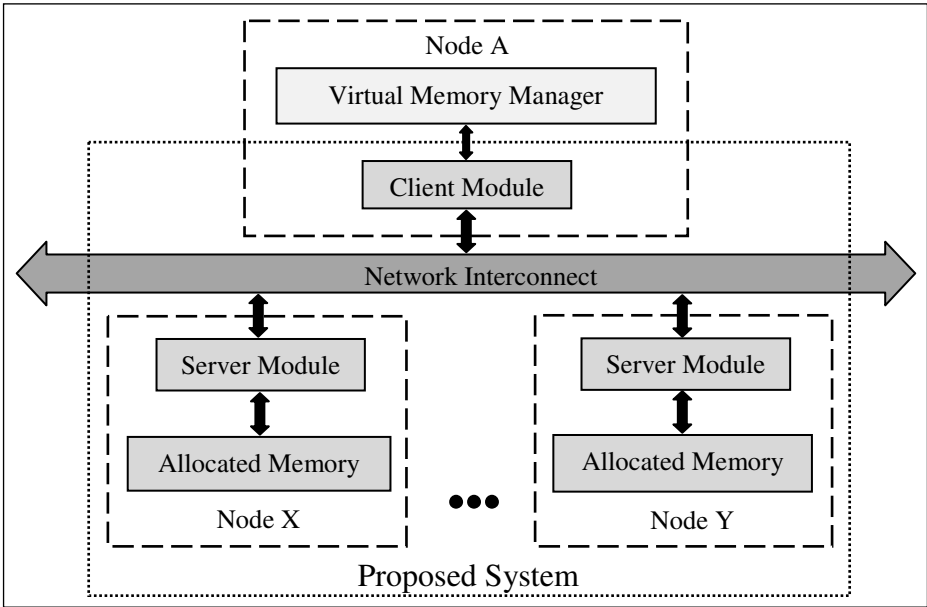


Fig. 1. Block Diagram of the System

Request queue is the linked list of I/O requests pending with the client. Field worker thread is a kernel thread to process the pending requests in the request queue.

Client Layer 1 is implemented as the *device_make_request(DMR)* module. Kernel invokes this module to submit the block I/O structure, named as *bio*, to the device driver (client). When kernel calls *DMR* module, it maps the *bio* (page) to the appropriate server (among the servers in the *Server List*) using *Swap Map*, and computes the offset within the *Swap Block* to have exact memory location on the remote server. Further, it encapsulates the *bio* in its custom request structure and then appends the request structure in the request queue of the server, and in this way forwarding I/O request to Layer 2. Figure 2 shows the block diagram of the client module.

Client Layer 2 contains the *Server List*, its each node has a worker thread to process the pending requests in queue corresponding to the server. When swapping occurs there are multiple requests pending in the queue. Worker thread extracts the request from the queue and processes the request. It sends the suitable command (read or write) to the server module and transfers the data.

Mapping of pages is done in such a way that, load is evenly distributed among the worker threads in Layer 2. If device consists of n Swap Blocks, then any set of n consecutive pages are mapped onto different Swap Blocks. Further, the layered structure of the client allows the *DMR* module to return, just after submitting the *bio* to Layer 2, rather being blocked till the I/O request is completed. Thus it takes very little time to submit *bio* as compared to the time taken by Layer 2 to complete the I/O over network, and hence because of this, large number of requests are appended in the request queue of each server, which in turn allows the client to merge many consecutive requests and process multiple of them simultaneously with its worker threads.

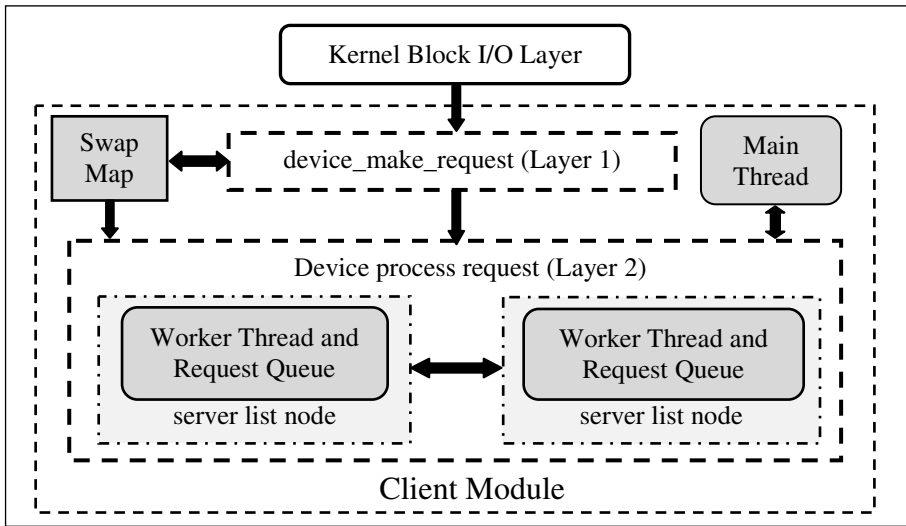


Fig. 2. Block Diagram of Client Module

In the implementation, we have bypassed the I/O scheduler layer and directly registered the *DMR* module to the Block I/O layer. Schedulers reorder and merge the requests to enhance the disk performance. But in our case, it may happen that the merged request may have the pages that map to two different Servers, as our device is based on distributed memory. In order to enhance the performance, we merge the requests that are pending in the request queue of the servers at Layer 2 of our client, as all the requests in the queue belong to the same server. This reduces the overheads of communication protocol.

3.3 Server Module

The server module monitors local memory load and allocates free memory for the clients and frees the allocated memory when client requests for it. The server module comprises of two components. The first component is Worker Component, which consists of allocated *Swap Blocks* and a set of worker threads to fulfill the client's read or write request for the associated *Swap Blocks*. The second component is Manager Component, it manages the first component and receives the control messages (discussed in section 3.4 Communication Protocol) from the clients and takes the suitable action. Figure 3 shows the block diagram of the server module and the way components are interconnected with each other.

Worker Component has *Client Space List* and *Swap Block Map*. Each node of *Client Space List* has following fields: Client IP Address, worker thread, pointer to connected socket, and pointer to *Swap Block Map*. *Swap Block Map* maps the swap block number to the starting logical address of the *Swap Block*. Server Manager Component allocates memory and initializes node for each client to which space has been allocated and adds it in the *Client Space List*.

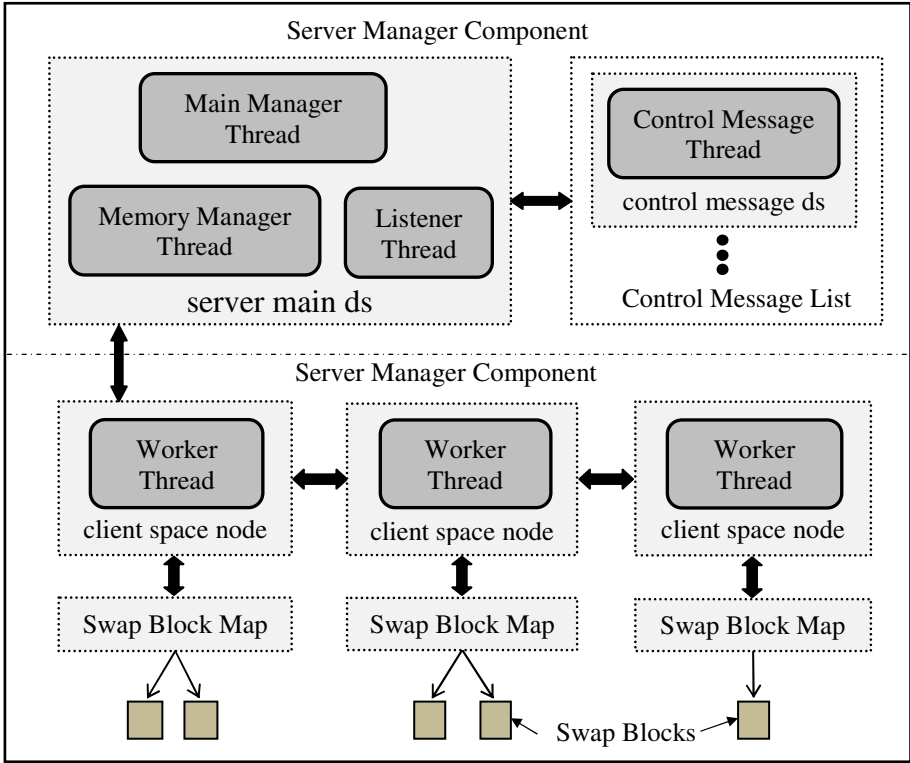


Fig. 3. Block Diagram of Server Module

Manager component consists of following: a Main Manager thread, a Listener thread, a Memory Manager thread, Control Message list, a memory status flag. The control message is request from client to allocate or free the memory.

Main Manager thread is invoked when Server Module is loaded, Main Manager initializes the data structures of the server module, and it starts the Listener and Memory Manager threads. The memory manager thread monitors the local memory and updates the memory status flag, for the main manager thread, at regular intervals. The listener thread listens on the well known port for any connection from the clients. When listener thread accepts a new connection, it initializes new *control message ds* structure and starts the control message thread. Control message thread communicates with the Client and receives the control message, it then updates the main manager thread that message has been received. When there is control message in the list the main manager thread processes the message, it takes the suitable action depending on the control command type (discussed in next section, 3.4 Communication Protocol).

3.4 Communication Protocol

Client and server modules communicate using TCP/IP protocol. Client and Server communicate by sending Request and Reply control messages. Request message

mainly consists of control command type and other information specific to control command. Reply message mainly consists of error code, which describes the status of the control command. The following are the types of control commands:

- *Alloc Block*: This is sent by client to server, requesting it to allocate the *Swap Block*.
- *Disconnect*: This control command is sent by client to the server, requesting it to free all the allocated *Swap Blocks* and close the connection with it.
- *Read*: This control command is sent by the client to the server, to send n bytes of data from specified swap block, starting from given offset.
- *Write*: This control command is sent by the client to the server, to receive n bytes of data in the specified swap block, starting from given offset.

When main manager thread receives *Alloc_Block* command, it first checks the memory status flag. If the free memory is available, it allocates *Swap Block*. It initializes new *client space node* structure for the client. It starts the worker thread and updates other fields. It initializes the *Swap Block Map* (*swap block map* structure) with the swap block number and its logical address. Finally, it adds the *client space node* in the Client Space List.

When main manager thread receives *Disconnect* message, it frees all the *Swap Blocks* allocated to the client, stops the worker thread and closes the socket. It removes the client's *client space node* from the Client Space list.

Read and *Write* control commands are not processed by main manager thread, they are used for data transfer after the pseudo block device is being registered. Client side worker thread sends *Read* or *Write* control command to Server side worker thread to receive or send the data respectively.

4 Experimentation and Results

The execution times of the test programs needing very large memory have been used to compare the performance of the proposed network swap-device with the disk based swap. Section 4.1 describes the experimental setup used to perform the tests.

4.1 Experimental Setup

The experiments to compare swapping to disk and NRBD are conducted on a cluster of 20 nodes. A node has two quad core 3.0 GHz processors with 6 MB L2 cache and 32GB physical memory, InfiniBand and Gigabit Ethernet networks and a 7200 rpm 500 GB hard disk with a maximum transfer rate of 3 Gb/s. The Operating System is Linux and version 2.6.18.

We have setup a network RAM based swap device on two nodes, that is, both having NRBD client module loaded on them and each one them connected to the set of four nodes, running the NRBD server module (total eight server nodes). Each server allocates 4 GB of memory for the client, creating a network ram block device (NRBD) of 16 GB. We have 24 GB Disk swap space on each node.

4.2 Job Description and Performance Results

In order to evaluate the performance of the device in different situations, we selected three different types of workloads. We have written two of them, to test the devices under different memory access patterns, and the third workload is HPLinpack benchmark. The description of the workloads is as follows:

a) Job1 performs sequential writes to and reads from a large chunk of memory; it dynamically allocates large chunk of memory (which exceeds the free memory available) and then sequentially writes into it randomly generated integer values and then read them back in temporary variable. The idea was to observe the performance of contiguous access in swap device. In our implementation the logically contiguous swap device is spread across memory from different nodes. In this, network bandwidth and latency put a definitive penalty.

b) Job2 performs random writes/reads on a large chunk of memory, which in turn causes random access to swap device.

c) HPLinpack benchmark solves dense system of linear equations and is used to rank Top 500 supercomputer sites. It well known benchmark and simulates the memory intensive parallel application.

To compare the NRBD with disk based swap the average of execution time of 10 runs is taken into consideration. Workloads were run using the NRBD as the swap device under two interconnects – InfiniBand (IPoIB) and Gigabit Ethernet (GigE). We have recorded the baseline timing for the Job1 and Job2 with sufficient memory available.

For the HPLinpack runs, we used full 32 GB of RAM. We kept its problem size sufficiently large to cause swapping on the node with of 32 GB RAM. We ran the HPLinpack benchmark on two clients simultaneously.

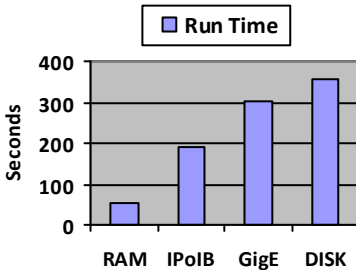


Fig. 4. Job 1 Performance Results

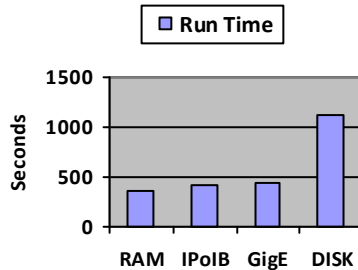


Fig. 5. Job 2 Performance Results

Figure 4 and Figure 5 gives the pictorial view of the results. In case of Job 1 we have got 1.85 times speedup with NRBD-IPoIB and 1.18 times speedup with NRBD-GigE as compared to disk. For Job 2 we have got 2.7 times speedup with NRBD-IPoIB and 2.5 times speedup with NRBD-GigE as compared to disk.

Figure 6 gives the pictorial view of HPLinpack results. For HPLinpack we have got very encouraging 5.4 times speedup with NRBD-IPoIB and 4.5 times speedup with NRBD-GigE as compared to disk.

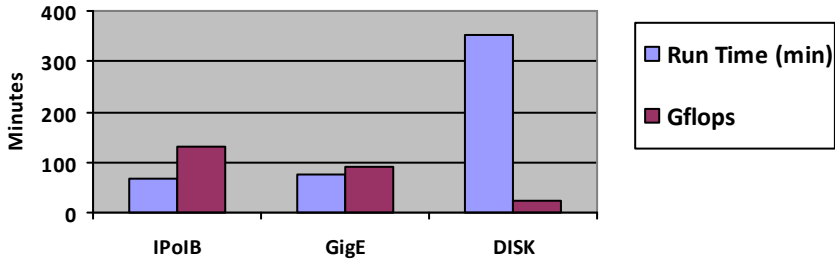


Fig. 6. HPLinpack Performance

5 Conclusions and Future Work

We have designed, developed and deployed remote memory based network swap over both InfiniBand and Gigabit Ethernet networks. Through experimentation we have demonstrated that significant benefits can be derived in improving turnaround time of the jobs. With our implementation using additional memory for the purpose of executing large program would be as simple as including a few machines with memory on the network. Furthermore our current work can be used for sizing the memory requirements of a given application and their number of instances to get an acceptable turnaround time by factoring in network delays.

Further, we wish to pursue research work on finding the swap block replacement policies and incorporate redundancy issues to address failure of a participating node. With the increase in speeds and reduced latencies of Network interconnects, utilizing remote memory for swapping will be increasingly deployed in clusters to gain performance benefits. Swapping on to remote memory itself might become the norm with a storage device (Like a SAN box) may be used occasionally as a backing store for storing the snapshots of the network based swap utility to address failure of participating node/nodes.

Acknowledgements

We would like to express our sincere gratitude to R. S. Mundada, Kislay Bhatt, D. D. Sonvane, Vaibhav Kumar, Vibhuti Duggal, Urvashi Karnani of Parallel Processing Group, Computer Division, BARC, for their insightful and constructive suggestions throughout the research work. We are thankful to A.G. Apte, Head, Computer Division, BARC, for providing us with the opportunity to undertake this project.

References

1. Acharya, A., Setia, S.: Availability and Utility of Idle Memory on Workstation Clusters. In: ACM SIGMETRICS Conference on Measuring and Modeling of Computer Systems, pp. 35–46 (May1999)

2. Arpaci, R.H., Dusseau, A.C., Vahdat, A.M., Liu, L.T., Anderson, T.E., Patterson, D.A.: The Interaction of Parallel and Sequential Workloads on a Network of Workstations. In: ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems, pp. 267–278 (1995)
3. Feeley, M.J., Morgan, W.E., Pighin, F.H., Karlin, A.R., Levy, H.M., Thekkath, C.A.: Implementing Global Memory Management in a Workstation Cluster. In: 15th ACM Symposium on Operating Systems Principles (December 1995)
4. Markatos, E.P., Dramitinos, G.: Implementation of a Reliable Remote Memory Pager. In: USENIX 1996 Annual Technical Conference (1996)
5. Bernard, G., Hamma, S.: Remote Memory Paging in Networks of Workstations. In: SUUG 1994 Conference (April 1994)
6. Anderson, E., Neeffe, J.: An Exploration of Network RAM. Technical Report CSD-98-1000, UC Berkley (1998)
7. Liang, S., Noronha, R., Panda, D.K.: Swapping to remote memory over InfiniBand: an approach using a high performance network block device. In: IEEE Cluster Computing (2005)
8. Newhall, T., Finney, S., Ganchev, K., Spiegel, M.: Nswap: a network swapping module for linux clusters. In: Proc. Euro-Par 2003 International Conference on Parallel and Distributed Computing (2003)
9. Newhall, T., Amato, D., Pshenichkin, A.: Reliable Adaptable Network RAM. In: IEEE International Conference on Cluster Computing (2008)

Collaborative Alert in a Reputation System to Alleviate Colluding Packet Droppers in Mobile Ad Hoc Networks

K. Gopalakrishnan and V. Rhymend Uthariaraj

Ramanujan Computing Centre, College of Engineering Guindy,
Anna University Chennai, Chennai – 600 025, Tamil Nadu, India
mrkrishauc@yahoo.in, rhymend@annauniv.edu

Abstract. The nature of the ad hoc network seems to have a promising future in real world and vast researches are going on to make the network more secure in an open wireless environment. The misbehaving nodes in ad hoc network results in degradation of overall network throughput and creates difficulty in finding route between nodes. Thus the collaborative nature of ad hoc network seems to be endangered due to the presence of misbehaving nodes and even get worse when the misbehaving node colludes to misbehave. This paper addresses a colluding packet dropping misbehavior and proposes a collaborative alert mechanism in a reputation system to alleviate it. The simulation result shows that the proposed system increases overall network throughput, reduces the malicious drop and false detection when compared to existing system and defense less scenario.

Keywords: Routing Security, Collaborative Alert, Reputation System, Colluding Packet Droppers, Ad Hoc Networks.

1 Introduction

Mobile Ad Hoc Networks (MANETs) is a collection of mobile nodes that communicates with each other by not depending on the preexisting infrastructure and centralized administration. A routing protocol is used to discover correct and efficient route between a pair of nodes so that messages may be delivered in a timely manner. The lack of preexisting infrastructure makes each node in the network to function as routers which discover and maintain routes to other nodes in the network. Ad hoc network maximizes the total network throughput by using all available nodes for routing and forwarding. The more nodes that participate in the routing process results in greater the aggregate bandwidth, shorter routing paths and minimizes the network partition. The lack of centralized administration and the nature of ad hoc networks pose a threat to the routing process [11]. A node agrees to forward packets on behalf of other nodes during the route discovery phase but failed to do so due to malicious or non-malicious behavior [13]. The non-malicious packet dropping exists due to network congestion, mobility and node malfunction. When the malicious node

colludes to mischief then it further increases the complexity in discovering routes and also results in frequent network partitioning and performance degradation. This paper addresses the colluding packet dropping misbehavior and proposes a novel solution to alleviate it. The rest of the paper is organized as follows. The section 2 describes about the related works. The proposed work is described in section 3. In section 4 and 5 the simulation study and the results are discussed. Finally, section 6 concludes the work and insights about the future work.

2 Related Works

Marti et al. [5] proposed a scheme which contains two components namely *Watchdog and Pathrater* in each and every node to detect and mitigate the routing misbehavior in mobile ad hoc networks. The nodes operate in promiscuous mode where in the watchdog maintain a buffer of recently sent packets and compare each overheard packet with the packet in the buffer. If there is a match then the packet in the buffer is removed else if the packet remained in the buffer for longer than a certain time out then the watchdog increments a failure tally for the node responsible for forwarding the packet. If the tally exceeds certain threshold limit then the node is identified as misbehaving. The pathrater combines the knowledge of misbehaving nodes with link reliability data to pick a route that is most likely to be reliable. This approach does not punish misbehaving nodes that do not cooperate and also relieves them of the burden of forwarding packets for other nodes. Buchegger et al. [7] proposed a protocol called *CONFIDANT (Cooperation Of Nodes: Fairness In Dynamic Ad-hoc NeTwork)* based on selective altruism and utilitarianism to detect and isolate misbehaving nodes. The protocol adds a trust manager and a reputation system to the watchdog and pathrater scheme. The trust manager evaluates the events reported by the watchdog and the reputation system maintains a blacklist of nodes at each node and shares with all other nodes that are in their friends list. Trust relationships and routing decisions are based on experienced, observed, or reported routing and forwarding behavior of other nodes. Michiardi et al. [8] proposed a mechanism called *CORE (Collaborative REputation mechanism)* to enforce node cooperation in mobile ad hoc network. CORE stimulates node cooperation based on collaborative monitoring and a reputation mechanism. The reputation metric is computed based on data monitored by the local entity and some information provided by the other nodes involved in each operation. The nodes with good reputation can utilize the network resources where as the node with bad reputation are gradually excluded from the network. Bansal et al. [10] proposed a reputation mechanism termed as *OCEAN (Observation-based Cooperation Enforcement in Ad hoc Networks)* based on direct observation experienced by a node from its neighbors. The routing decisions are based on direct observation of neighboring nodes behavior and it completely disallows the exchange of second hand reputation. Hu et al. [12] proposed a scheme called *LARS - A Locally Aware Reputation System* for Mobile Ad Hoc Networks for which the reputation of nodes is derived by using direct observation. When a selfish node is identified then its k-hop neighbors become aware of the selfishness, where k is a parameter which is adaptive

to the security requirement of the network. In order to avoid false accusation and the associated trust issues, conviction of the selfish node is valid only if m different neighbors accuse, where $m - 1$ is an upper bound on the number of malicious nodes in the neighborhood. The success of this scheme relies on the critical selection of value for m .

3 Collaborative Alert in a Reputation System

The proposed *Collaborative Alert in a Reputation System (CARS)* consists of three main components a monitor to detect the packet dropping misbehavior, reputation system to maintain the trust value for the neighborhood nodes and a path manager to maintain the routes without containing packet droppers in it as shown in Fig. 1.

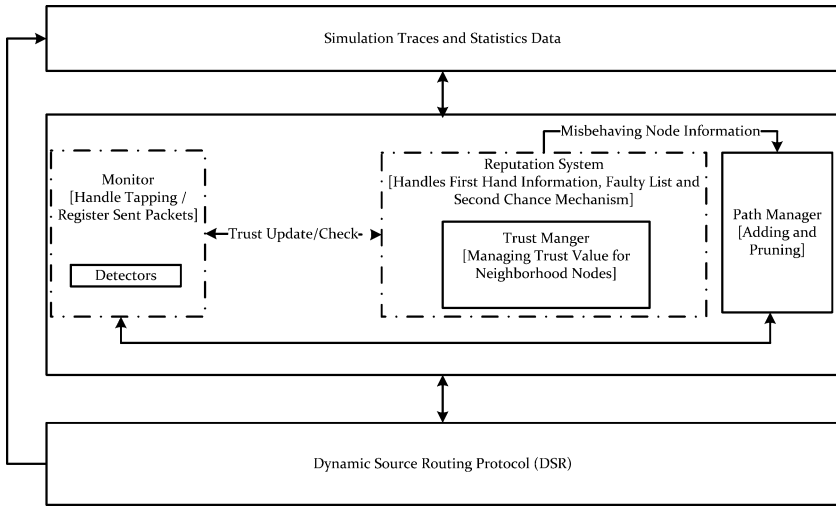


Fig. 1. Functional Block Diagram of CARS

These components are added as an add-on into the existing routing functionality of *Dynamic Source Routing (DSR)* protocol. This enables each node in the network to execute this add-on functionality along with the usual routing protocol operations. Whenever a node overhears a packet from the neighboring node for the first time then the neighboring node information is stored in the *Neighbor Connectivity List (NCL)* along with the timestamp at which the packet is overheard and its trust value is initialized into 0. The timestamp and the trust value are updated for the subsequent packet overhearing from the neighboring node. The monitor component is responsible for tapping and registering the sent packets. It has an internal component called detectors which are used to identify the different kinds of packet dropping misbehavior as described in sub section

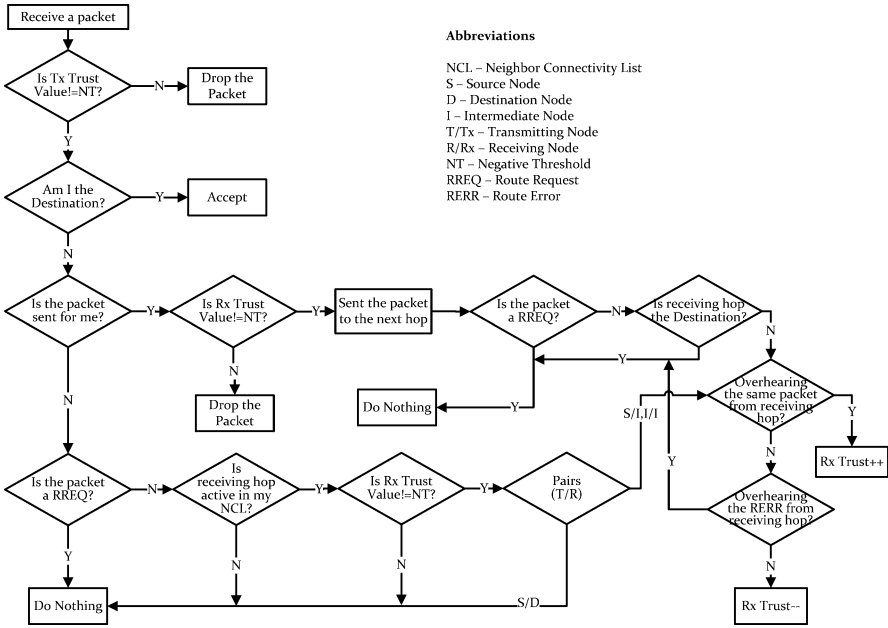
4.1. The reputation system handles the first hand information received from the monitor and maintains the trust value for the neighborhood nodes accordingly with the help of its internal component called trust manager. Once a nodes reputation value reaches the *Negative Threshold* limit then it will be added into the faulty list and any traffic to and from the misbehaving node will be rejected. As soon as a node is added into the faulty list a second chance timer will be initiated for that node.

The misbehaving node information is also communicated to the path manger in order to prune the routes which have the misbehaving link in it and also an explicit route error packet will be sent to the source of the packet to inform about the misbehaving link. Once the source or an intermediate node receives an explicit route error packet then it checks whether it is originated by the source of the misbehaving link or from the neighborhood of the misbehaving node. The route which contains the misbehaving link is pruned if the packet is originated by the source of the misbehaving link else routes containing the destination of the misbehaving link will be pruned from both primary and secondary route cache. When the second chance timer of the misbehaving node reaches 100s then the node is removed from the faulty list and reintroduced into the network by considering it to be useful again after reducing its trust value by half. The reason for not resetting the trust value of the reintroduced node to 0 is that the node might still continue to misbehave. If it continues to misbehave then it will be detected soon.

3.1 Packet Monitoring and Trust Evaluation of CARS

The procedure for packet monitoring and trust evaluation of the proposed system is shown in Fig. 2. Whenever a node receives a packet it checks the trust value of the transmitting node. If it is not equal to *Negative Threshold* then it further checks whether it is a destination or not else drops the packet.

It accepts the packet if it is a destination node else it checks whether it's a forwarding or an overhearing node. If it is a forwarding node it checks whether the receiving hop trust value is not equal to *Negative Threshold*, if so it forwards the packet to the next hop else drops the packet. Further it checks whether the forwarded packet is a *RREQ* or the receiving hop is a destination. If so the procedure ends else it waits to overhear the same packet forwarded by the receiving hop. The trust value of the receiving hop is incremented by 1 if it overhears the packet else it waits until the packet timeout period to overhear the *RERR* packet. If it does not overhear the *RERR* packet then the trust value of the receiving hop is decremented by 2 else the procedure ends. On the other hand, if it is an overhearing node then it checks whether the receiving hop is active in its *NCL* and its trust value is not equal to *Negative Threshold*. If so it checks the relation between transmitting and receiving hop else the procedure ends. The procedure ends if the relation of receiving hop is a destination else it waits to overhear the same packet from the receiving hop. If it overhears the packet then the trust value of the receiving hop is incremented by 1 else it waits for the *RERR* packet. The procedure ends if it overhears the *RERR*



Abbreviations
 NCL - Neighbor Connectivity List
 S - Source Node
 D - Destination Node
 I - Intermediate Node
 T/Tx - Transmitting Node
 R/Rx - Receiving Node
 NT - Negative Threshold
 RREQ - Route Request
 RERR - Route Error

Fig. 2. Packet Monitoring and Trust Evaluation of CARS

packet within packet timeout period else the trust value of the receiving hop is decremented by 2. The neighboring node is considered to be active if any kind of packet is already overheard from it within 3000ms at the time of checking. The RREQ packet is not monitored because it can be dropped due to network operations [9].

3.2 Colluding Packet Dropping Misbehavior

As shown in Fig. 3, the solid circle represents a node and a solid line between them shows that the nodes are within the communication range of each other. Assume that the source node *S* communicates with the destination node *D* via the intermediate nodes $I_1 \rightarrow I_2 \rightarrow I_3$ and the nodes I_2 and I_3 colludes to misbehave. If I_3 drops the packet then the previous hop I_2 will not monitor and report to the source of the packet about this spiteful behavior because it colludes to mischief with node I_3 .

In this scenario the neighboring nodes N_1 and N_2 are within the transmission range of I_3 so it can identify spiteful behavior and report to the source of the packet about this misbehaving link. Once the trust value of the misbehaving node I_3 reaches the *Negative Threshold* in N_1 or N_2 then node I_3 was added into their faulty list and also an explicit route error packet has been generated and sent to the source of the packet to inform about this misbehaving link $I_2 \rightarrow I_3$. The source route of an explicit route error packet should not contain node I_2 in it because it colludes to misbehave with I_3 . Once the source or an intermediate node receives

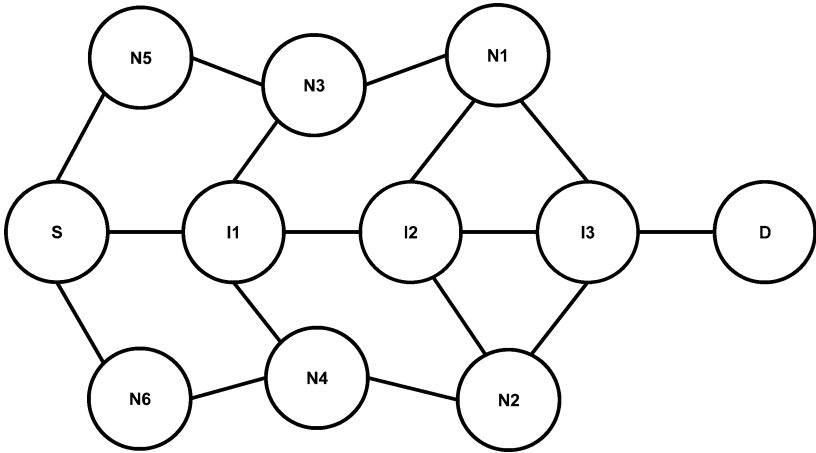


Fig. 3. Colluding Packet Dropping Misbehavior

an explicit route error packet then it will prune the routes from both the primary and secondary cache which have the misbehaving node I_3 in it.

4 Simulation Study

The proposed system was implemented in *ns-2.34* as an add-on to the *DSR* routing protocol. The *DSR* is an on demand source routing protocol which supports for promiscuous listening in order to overhear the neighboring nodes transmission and updates its trust value accordingly. Further, the source routing allows packet routing to be trivially loop-free, avoids the need for up-to-date routing information in the intermediate nodes through which packets are forwarded, and allows nodes forwarding or overhearing packets to cache the routing information in them for their own future use [2]. In simulation two different kinds of mobility models were used to mimic the real world movement of the mobile nodes and evaluated the performance of the proposed system in each of them separately. The first one is a *Random Waypoint (RWP)* mobility model based on *Entity (E)* mobility model in which the mobile node movements are independent of each other and the other one is a *Reference Point Group Mobility (RPGM)* Model based on *Group (G)* mobility model in which the mobile nodes move as a group [4]. The *Random Waypoint* mobility model is based on *CMU Monarch v2* implementation and *Reference Point Group Mobility* model is based on the implementation of [6].

There exists multiple group of mobile nodes and each group work towards different goal but there exists a communication between groups as described in [1], [3] so the group mobility model utilizes both inter and intra group *CBR* traffic patterns to evaluate the proposed system under the group mobility scenario. Each node is assigned an initial value of *Energy (E)* by using an uniform distribution function in the interval $(E_i - 3J, E_i + 3J)$ where the energy is expressed

Table 1. Simulation Parameters

Parameter	Value
Simulation Area	900 m x 900 m
Simulation Time	900 s
Number of Nodes	50
Number of Groups	5
Nodes Per Group	10
Reference Point Separation	100
Node Separation from Reference Point	50
Propagation Model	Two-Ray Ground Reflection
Antenna	Omni Directional Antenna
RXThresh	3.65262e-10
Packet Timeout	0.5 s
Transmission Range	250 m
Transmission Power	0.281838W
Reception Power	0.281838W
Traffic Type	CBR (UDP)
Maximum Connections	15
Payload Size	512 bytes
Seed Value	1-20
Negative Threshold	-40
Positive Threshold	40

in *Joules* (J) and the initial energy $E_i = 500J$. The consequence of this choice is that every node will run out of energy at different times in the simulation, which adds a degree of randomness to the simulation for evaluating the performance of the proposed scheme. The simulation parameters that were used in the simulation are shown in Table 1.

4.1 Modeling the Misbehavior

The proposed system was simulated by introducing three different kinds of packet dropping misbehavior as listed below

1. Packet Dropping Type 1 - These nodes participate in the *DSR* Route Discovery and Route Maintenance phases, but refuse to forward data packets on behalf of other nodes (which are usually much larger than the routing control packets)
2. Packet Dropping Type 2 - These nodes participate in neither the Route Discovery phase, nor forwarding data packets. They only use their energy for their own packet transmission
3. Packet Dropping Type 3 - These nodes behave differently based on their energy levels. When the energy lies between full energy E and a threshold T_1 , the node behaves properly. On the other hand if an energy level lies between T_1 and another lower threshold T_2 then it behaves like a node of

Packet Dropping Type 1. Finally, for an energy level lower than T_2 , it behaves like a node of Packet Dropping Type 2. The relationship between T_1 , T_2 and E is $T_2 < T_1 < E$

The node behavior has been added as a node definition type in the *ns2* node model. The syntax that is used to define the node configuration has been enhanced with a new optional feature that allows selecting the packet dropping model among three possible choices. It is also necessary to modify the *DSR* routing protocol implemented in *ns2* because the networking functions (routing and packet forwarding) are overridden by the routing protocol selected in the node configuration. The modified version of the *DSR* routing protocol checks the current node configuration and depending on the packet drop model used for that node, it decides whether to execute the networking functions or not.

4.2 Performance Metrics

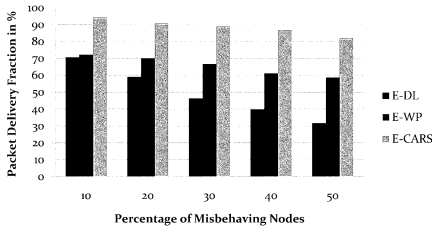
The performance of the proposed system has been measured by using the following parameters

1. Packet Delivery Fraction (%) - The packet delivery fraction is measured in terms of the ratio of the data packets delivered to the destinations to those generated by the *Constant Bit Rate (CBR)* sources
2. Normalized Routing Load (Packets) - The number of routing packets transmitted per data packet delivered at the destination. Each hop-wise transmission of a routing packet is counted as one transmission
3. Average End-End Delay (Seconds) - This includes all possible delays caused by buffering during route discovery latency, queuing at the interface queue, retransmission delays at the *MAC*, propagation and transfer times
4. Average Energy Dissipation (Joules) - The average amount of network energy dissipated over the simulation period
5. Malicious Drop (Packets) - The total number of packets dropped by the different kind of packet droppers
6. False Detection (%) - The percentage of nodes detected falsely as a misbehaving node over the simulation period

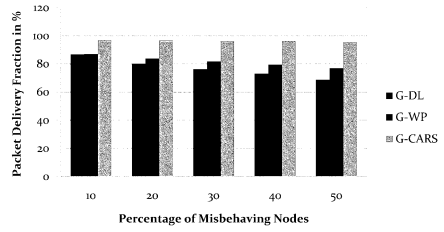
The measurements of the network performance were made using a script that parses and analyzes the trace file output generated from the simulation. The trace file provides information about a set of defined events that occurred in the simulation such as medium access control layer events, routing layer events and agent level events.

5 Results and Discussions

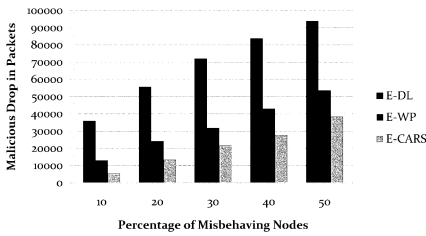
The simulation results of the proposed system were compared with *Defense Less (DL)* scenario and the existing scheme *Watchdog Pathrater (WP)* [5]. This paper calculates a 95% confidence interval for the unknown mean and plots the



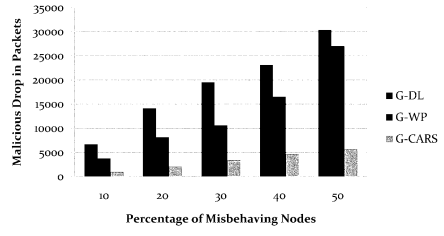
(a) Entity Mobility Scenario



(b) Group Mobility Scenario

Fig. 4. Packet Delivery Fraction in %

(a) Entity Mobility Scenario



(b) Group Mobility Scenario

Fig. 5. Malicious Drop in Packets

confidence intervals on the graphs. The packet delivery fraction of *CARS* was increased by 24-51% and 11-27% when compared to *DL* scenario, 21-26% and 11-19% when compared to *WP* scheme under both entity and group mobility scenario respectively as shown in a,b of Fig. 4.

As shown in a,b of Fig. 5, the malicious drop of *CARS* has been decreased by 60-85% and 82-86% when compared to *DL* scenario, 29-58% and 68-79% when compared to *WP* scheme with respect to both entity and group mobility scenario respectively. The false detection of *CARS* was decreased from 44-61% and 50-77% when compared to *WP* scheme under both entity and group mobility scenario respectively as shown in a,b of Fig. 6. As shown in a,b of Fig. 7, the normalized routing load of *CARS* has been decreased by 20-35% and 32-63% when compared to the *WP* scheme under entity and group mobility scenario respectively. Since the average energy dissipation is directly proportional to the overall network throughput. As shown in a,b of Fig. 8, the average energy dissipation of *CARS* was increased by 6-9% and 5-6% when compared to *WP* scheme under entity and group mobility scenario respectively. The average end-end delay has been decreased by 3-46% and 14-70% when compared to *WP* scheme under entity and group mobility scenario respectively as shown in a,b of Fig. 9. The result shows that the proposed system performs better when compared to

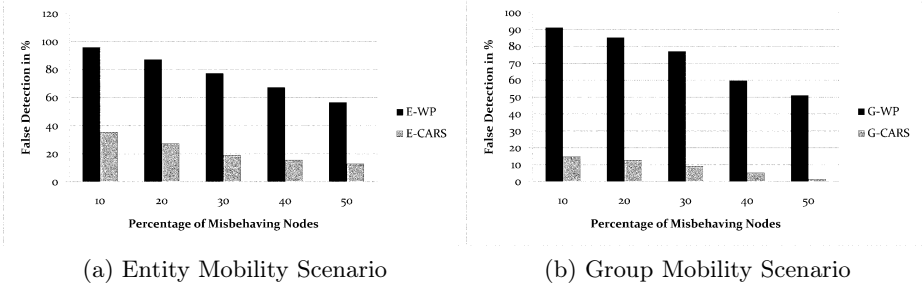


Fig. 6. False Detection in %

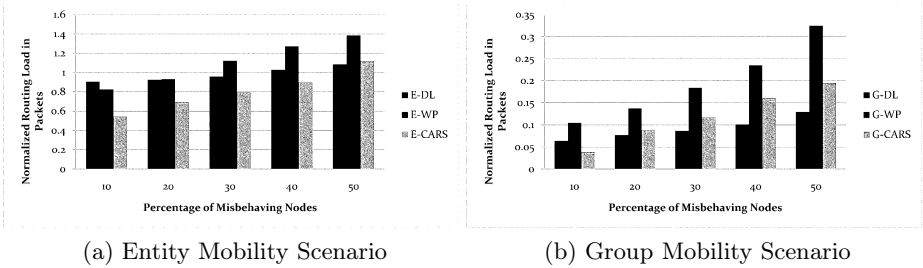


Fig. 7. Normalized Routing Load in Packets

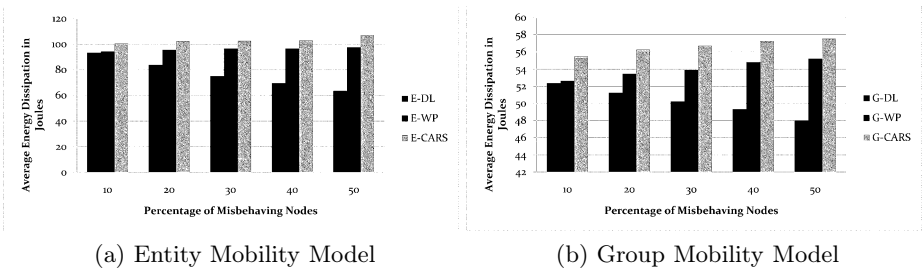


Fig. 8. Average Energy Dissipation in Joules

the Watchdog Pathrater scheme. The timely generation of an explicit route error generation reduces the false detection and in turn increases the overall network throughput. The strong malicious traffic rejection results in reduction in control overhead and end-end delay of per packet delivered to the destination.

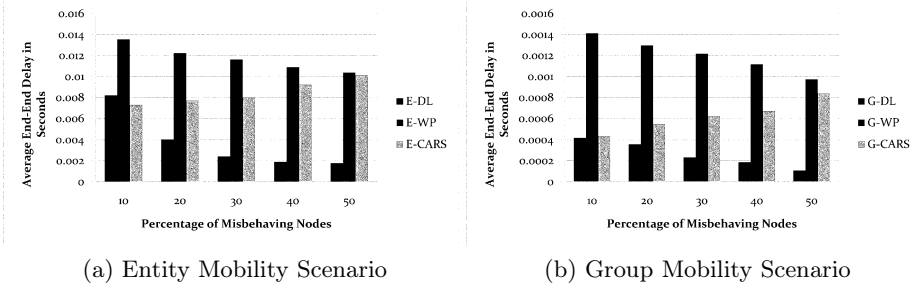


Fig. 9. Average End-End Delay in Seconds

6 Conclusion and Future Work

The simulation results of the proposed system shows that the overall network throughput has been greatly improved and the percentage of malicious packet drop also reduced due to efficient detection and isolation of packet droppers. Further, the timely generation of an explicit route error packet to inform the source of the packet about the misbehaving link combined with the forward traffic rejection reduces the percentage of false detection when compared to the existing scheme. The proposed system was immune to colluding packet dropping misbehavior because of an explicit route error generated by the neighboring nodes and also other kind of overhearing technique drawbacks due to the efficient monitoring and detection of neighboring nodes. In future work, more kind of misbehaving nodes will be considered and also the faulty list will be shared with the rest of the nodes in the network in order to mitigate the misbehaving nodes without incurring additional control overhead.

Acknowledgment. The first author would like to acknowledge the support received from *CSIR-HRDG (Council of Scientific and Industrial Research-Human Resource Development Group)*, India, through Senior Research Fellowship.

References

1. Hong, X., Gerla, M., Pei, G., Chiang, C.: A Group Mobility Model for Ad Hoc Wireless Networks. In: 2nd ACM International Workshop on Modeling and Simulation of Wireless and Mobile Systems, pp. 53–60. ACM, Seattle (1999)
2. Johnson, D.B., Maltz, D.A., Broch, J.: The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks. Internet Draft, The Internet Engineering Task Force (1999)
3. Liang, B., Haas, Z.: Predictive Distance-Based Mobility Management for PCS Networks. In: 18th Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 3, pp. 1377–1384. IEEE Computer Society, New York (1999)

4. Hong, X., Gerla, M., Pei, G., Chiang, C.: A Wireless Hierarchical Routing Protocol with Group Mobility. In: IEEE Wireless Communications and Networking Conference, pp. 1536–1540. IEEE, New Orleans (1999)
5. Marti, S., Giuli, T.J., Lai, K., Baker, M.: Mitigating Routing Misbehavior in Mobile Ad Hoc Networks. In: 6th International Conference on Mobile Computing and Networking, pp. 255–265. ACM, Boston (2000)
6. Camp, T., Boleng, J., Davies, V.: A Survey of Mobility Models for Ad Hoc Network Research. *J. Wireless Communication and Mobile Computing* 2, 483–502 (2002)
7. Buchegger, S., Le Boudec, J.Y.: Performance Analysis of the CONFIDANT Protocol (Cooperation Of Nodes: Fairness In Dynamic Ad-hoc NeTworks). In: IEEE/ACM Symposium on Mobile Ad Hoc Networking and Computing, pp. 226–236. ACM, Lausanne (2002)
8. Michiardi, P., Molva, R.: CORE: A Collaborative REputation Mechanism to enforce node cooperation in Mobile Ad hoc Networks. In: 6th Joint Working Conference on Communications and Multimedia Security, vol. 228, pp. 107–121. Kluwer, Portoroz (2002)
9. Tseng, Y.C., Ni, S.Y., Chen, Y.S., Sheu, J.P.: The Broadcast Storm Problem in a Mobile Ad Hoc Network. *J. Wireless Networks* 8, 153–167 (2002)
10. Bansal, S., Baker, M.: Observation-based Cooperation Enforcement in Ad hoc Networks. Technical Report, Stanford University (2003)
11. Yau, P., Mitchell, C.J.: Security Vulnerabilities in Ad Hoc Networks. In: The Seventh International Symposium on Communication Theory and Applications, pp. 99–104. HW Communications Ltd, Ambleside (2003)
12. Hu, J., Burmester, M.: LARS A Locally Aware Reputation System for Mobile Ad Hoc Networks. In: 44th Annual Southeast Regional Conference, pp. 119–123. ACM, Melbourne (2006)
13. Gopalakrishnan, K., Rhymend Uthariaraj, V.: Scenario based Evaluation of the Impact of Misbehaving Nodes in Mobile Ad Hoc Networks. In: 1st IEEE International Conference on Advanced Computing, pp. 45–50. IEEE Computer Society, Chennai (2009)

Secure Service Discovery Protocols for Ad Hoc Networks

Haitham Elwahsh, Mohamed Hashem, and Mohamed Amin

King Saud University, Ain shams University, Minoufiya University

Haitham.elwahsh@gmail.com, {mhashem100,mohamed_amin110}@yahoo.com

Abstract. Ad-hoc networks, mobile devices communicate via wireless links without the aid of any fixed networking infrastructure. These devices must be able to discover services dynamically and share them safely, taking into account ad-hoc networks requirements such as limited processing and communication power, decentralized management, and dynamic network topology, among others. Legacy solutions fail in addressing these requirements. In this paper, we propose a service discovery protocol with security features, the Secure Pervasive Discovery Protocol. SPDP is a fully distributed protocol in which services offered by devices can be discovered by others, without a central server. It is based on an anarchy trust model, which provides location of trusted services, as well as protection of confidential information, secure communications, or access control.

Keywords: ad-hoc networks, service discovery protocol, security.

1 Introduction

Recent advances in microelectronic and wireless technologies have fostered the proliferation of small devices with limited communication and processing power. They are what are known as “pervasive systems”. Personal Digital Assistants (PDAs) and mobile phones are the more “visible” of these kinds of devices, but there are many others that surround us, unobserved. For example, today most household appliances have embedded microprocessors. Each one of these small devices offers a specific service to the user, but thanks to their capacity for communication, in the near future they will be able to collaborate with each other to build up more complex services. In order to achieve this, devices in such “ad-hoc” networks should dynamically discover and share services between them when they are close enough. In ad-hoc networks composed of limited devices, it is very important to minimize the total number of transmissions, in order to reduce battery consumption of the devices. It is also important to implement mechanisms to detect, as soon as possible, both the availability and unavailability of services produced when a device joins or leaves the network. Security in these networks is also critical because there are many chances of misuse both from fraudulent servers and from misbehaving clients. In this paper, we propose a service discovery protocol with security features, the Secure Pervasive Discovery Protocol (SPDP). SPDP is a fully distributed protocol in which services offered by devices can be discovered by others, without a central server. It provides location of trusted services, as well as protection of confidential information, secure communications, identification

between devices, or access control, by forming a reliable ad-hoc network. The paper is organized as follows: section 2 enumerates the main service discovery protocols proposed so far in the literature, we will see that none of them adapts well to ad-hoc networks. Section 3 presents our secure pervasive discovery protocol, SPDP, with its application scenario, and description of the algorithm. In section 4 we describe the underlying trust model as security support. In section 5 we present the simulation results comparing SPDP with other services discovery protocols. Finally, we conclude with some conclusions and future work.

2 Related Works

Dynamic service discovery is not a new problem. There are several solutions proposed for fixed networks, with different levels of acceptance, like SLP [RFC2608, 1999] [1], Jini [Sun, 1999] [4] and Salutation [Miller and Pascoe, 2000] [5]. More recently, other service discovery protocols, specifically designed for ad-hoc networks, have been defined, some tied to a wireless technology (SDP for Bluetooth [SDP, 2001] [6], IAS for IrDA [IrDA, 1996]) [7], others that jointly deal with the problems of ad-hoc routing and service discovery (GSD [Chakraborty et al., 2002] [8], HSID [Oh et al., 2004]) [9], and others that work at the application layer of the protocol stack (DEAPspace [Nidd, 2001] [12], Konark [Helal et al., 2003] [13], and the post-query strategies [Barbeau and Kranakis, 2003]) [14]. Only a few protocols have built-in security, the most important are SSDS [Czerwinski et al., 1999] [16] and Splendor [Zhu et al., 2003]. However, these solutions cannot be directly applied to an ad-hoc network, because they were designed for and are more suitable for (fixed) wired networks. We see three main problems in the solutions enumerated: – First, many of them use a central server, such as SLP2, Jini and Salutation. It maintains the directory of services in the network and it is also a reliable entity upon which the security of the system is based. An ad-hoc network cannot rely upon having any single device permanently present in order to act as central server, and furthermore, maybe none of the devices present at any moment may be suitable to act as the server. – Secondly, the solutions that may work without a central server, like SSDP, are designed without considering the power constraints typical in wireless networks. They make an extensive use of multicast or broadcast transmissions which are almost costless in wired networks but are power hungry in wireless networks. – Thirdly, security issues are not well covered. SSDS provides security in enterprise environments but may not work in ad-hoc networks with mobile services. Splendor does not provide certificate revocation and trust models of PKIs. They both depend on trustworthy servers and they propose solutions which are provided at the IP level. Accepting that alternatives to the centralised approach are required, we consider two alternative approaches for distributing service announcements: – The “Push” solution, in which a device that offers a service sends unsolicited advertisements, and the other devices listen to these advertisements selecting those services they are interested in. – The “Pull” solution, in which a device requests a service when it needs it, and devices that offer that service answer the request, perhaps with third devices taking note of the reply for future use. In ad-hoc networks, it is very important to minimise the total number of transmissions, in order to reduce battery consumption. It is also important to implement

mechanisms to detect as soon as possible both the availability and unavailability of services produced when a device joins or leaves the network. These factors must be taken into account when selecting between a push solution and a pull solution. The DEAPspace algorithm is the only service discovery protocol, listed above, that tries to minimize the total number of transmissions. It uses a pure “push” solution and each device periodically broadcast its “world view” although none of them has to request a service.

3 SPDP: Secure Pervasive Discovery Protocol

In this paper we propose a new service discovery protocol, the Secure Pervasive Discovery Protocol (SPDP), which merges characteristics of both pull and push solutions to improve the performance of the protocol. Also, SPDP provides security based on an anarchy trust management model. Such trust management model does not require neither a central trusted server nor a hierarchical architecture, so it is suitable to overcome the challenges imposed by ad-hoc networks such as no central management, no strict security policies and highly dynamic nature (see section 4). The Secure Pervasive Discovery Protocol (SPDP) is intended to solve the problem of enumerating the services available in ad-hoc networks, composed of devices with limited transmission power, memory, processing power, etc. Legacy service discovery protocols use a centralized server that listens for broadcast or multicast announcements of available services at a known port address, and lists the relevant services in response to enquiries. The protocol we propose does away with the need for the central server. Ad-hoc networks cannot rely upon having any single device permanently present in order to act as central server, and further, none of the devices present at any moment may be suitable to act as the server. One of the key objectives of the SPDP is to minimize battery use in all devices. This means that the number of transmissions necessary to discover services should be reduced as much as possible. A device announces its services only when other devices request the service. Service announcements are broadcasted to all the devices in the network, all of which will get to know about the new service simultaneously at that moment, without having to actively query for it. In addition, SPDP allows sharing services safely, through an underlying trust management model between devices, which allows us to store service information from other “alleged” trusted service agents and later to use them if such information is really authentic and pright. Currently, the security support provided by service discovery protocols are focused on authentication, integrity, and confidentiality [RFC2608, 1999] [Czerwinski et al., 1999] [Zhu et al., 2003]. Even more, some of them include authorization services as part of the discovery [Zhu et al., 2003]. Such support is based on IPsec [Kent and Atkinson, 1998] or traditional PKI in the last case. However, these security services could be not necessary for the discovery, but they could cause energy and processing consumption. Protecting both energy and processing consumption is a very essential issue for devices with limited capabilities. So we have considered providing basic security services to prevent certain attacks (i.e. DoS, false announcements, and false services) and to avoid the sending of unnecessary messages. In the remainder of this section, we present the application scenario for SPDP and some considerations to be taken into account. Then, we will formally describe the algorithm used to implement it.

3.1 Application Scenario

Let's assume that there is an ad-hoc network, composed of D devices, each device offers S services, and expects to remain available in this network for T seconds. This time T is previously configured in the device, depending on its mobility characteristics. Each device has an SPDP User Agent (SPDP UA) and an SPDP Service Agent (SPDP SA). The SPDP UA is a process working on the user's behalf to search information about services offered in the network. The Service Agent SPDP (SPDP SA) is a process working to advertise services offered by the device. The SPDP SA always includes the availability time T of its device in its announcements. Each device has a cache associated which contains a list of the services that have been heard from the network. Each element e of the cache associated to the SPDP UA has three fields: the service description, the service lifetime and the service expiration time. The service expiration time is the time it is estimated the service will remain available. This time is calculated as the minimum of two values: the time the device has promised to remain available, and the time the server announced that the service would remain available. Entries remove themselves from the cache when their timeout elapses. With regard to security, each device handles a list of reliable devices and the trust degree associated with them. Trust helps devices to limit their cache size; services from untrusted devices are not stored in the cache. Depending on the trust degree, a device decides to store the service offered by a device on its cache. When the devices access services, devices with biggest trust degree are selected in the first place.

3.2 Algorithm Description

The SPDP has two mandatory messages: SPDP Service Request, which is used to send service announcements and SPDP Service Reply, which is used to answer a SPDP Service Request, announcing available services. SPDP has one optional message: SPDP Service Deregister, which is used to inform that a service is no longer available. Now, we will explain in detail how SPDP UA and SPDP SA use these primitives.

3.2.1 SPDP User Agent

When an application or the final user of the device needs a service of a certain type, it calls its SPDP UA. In order to support different application needs, in SPDP we have defined two kinds of queries:

- **one query–one response** (1/1): the application is interested in the service, not in which device offers it.
- **one query–multiple responses** (1/n): the application wants to discover all devices in the network offering the service. In this kind of query, we introduce a special type of service, named ALL, in order to allow an application to discover all available services of all types in the network.

3.2.2 SPSP Service Agent

The SPDP SA advertises services offered by the device. It has to process SPDP Service Request messages and to generate the corresponding SPDP Service Reply, if necessary. In order to minimize the number of transmissions, the SPDP SA takes into account the type of query made by the remote SPDP UA.

4 Evaluating the SPDP Protocol

In this section we present a performance evaluation study of SPDP in a ubiquitous computing environment. We compare our protocol with the theoretical distributed approaches, push and pull; because all the service discovery protocols defined in the literature are based on one of these approaches; and also we compare PDP with the service discovery protocol standard in Internet, SLP, and with UPnP's SSDP. This study was carried out through simulation using the well-known network simulator, NS-2. Our simulator is available in [Campo and Perea, 2004]. During the simulation, devices join the ubiquitous environment at random times, request and offer random services, and leave the network after a random time. The number of devices in the network varies over time, but its mean remains stationary. Random times follow exponential distributions, while random services follow uniform distributions. For simplicity we assume that each device offers just one service. The parameters of the simulation are: **the mean number of devices, the mean time they remain available in the network, the size of the caches, the mean time between service requests, and the total number of service types.** The results of interests are: the number of messages (the number of messages transmitted in the network normalized to the number of service request), the service discovery ratio (the ratio of services discovered to the total number of services available in the network) and the error ratio (the ratio of services discovered that were not available in the network to the total number of services discovered). Figure 6 shows the number of messages transmitted, the service discovery ratio and the error ratio, in a scenario with 20 devices, an average device life time ranging from 600 to 19200 seconds, a cache size of 100 entries, 5 different types of services, and each device requesting a random service every 60 seconds. The SPDP number of messages is quite under those obtained for SLP and for pull solutions, while keeping the same service discovery ratio and error rate of them.

(Fig. 2) shows the global power consumption in the same scenario as before. We see that, despite of using broadcast transmissions instead of unicast, and despite of sending bigger service requests (with the services already known from the cache), PDP achieves an important reduction in the power consumed, that is a reflection of the reduction of messages transmitted, and also of the lower energy cost associated with receiving than with transmitting. Only SSDP with an announcement period of "60 s" achieves less consumption, but at the cost of a higher error and lower service discovery percentages. Furthermore, as we will show later, PDP preserves the battery of the more limited devices, while the other protocols equally deplete the batteries of all devices. Regarding the delay in service discovery, it depends on the way the service discovery is done. In push mode protocols, the answer is obtained immediately from the cache. In directory-based protocols, the delay is the associated with transmitting a service request message to the directory, the processing time the directory needs to obtain the answer from its services database, and the transmission time associated with sending the reply. In pull mode protocols, as well as in Multicast DNS and PDP, the device broadcasts a service request (perhaps consulting first its local cache), and then it must wait for answers to come during a given period of time.

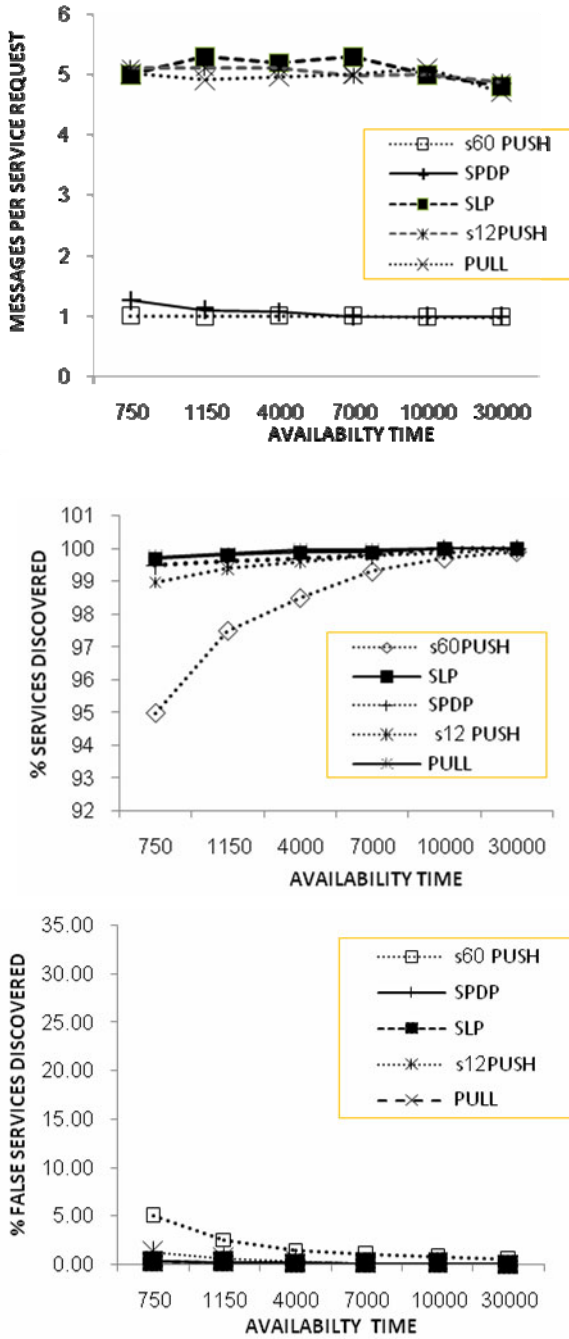


Fig. 1. Comparison of SPDP with others protocols

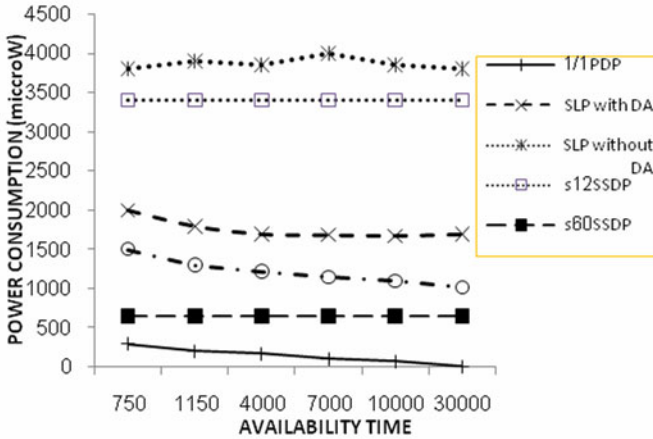


Fig. 2. Comparison of power consumption

Now, we will study the impact of the number of devices in the network and the cache size in the performance of PDP. A PDP with cache 0 is equivalent to a pull mode. (Fig. 3) shows that if the cache size is big enough, the number of messages transmitted remains constant, since all the services are already known and stored in the cache. For small cache sizes, when the number of devices equals the cache size, the number of messages starts growing linearly. For cache 0 (pull mode) the increment is always linear. Now, we will demonstrate how PDP takes into account device heterogeneity, achieving a reduction of traffic transmission (and so power consumption) in the more limited devices. (Fig. 4) shows the percentage of replies sent by each kind of devices depending on its availability time. We have considered a scenario with 40 devices in mean, with five different availability times: 500, 2500, 4500, 6500 and 9500 s, with about 20% of devices (in mean) of each type. The rest of parameters of the simulation are the same as before, except that the cache size for devices with availability time 500 and 2500 is 10 services, while for devices with availability time 4500 and 6500 is 40 services and for devices with availability time 9500 is 100 services. This way we simulate that devices that move more frequently (PDAs, mobile phones) have less memory than devices that move less frequently (laptops or desktop computers). In (Fig. 4) we see that devices with greater availability time answer more requests, preserving power consumption of devices with smaller availability in the figure sum up 70%, because in PDP some requests generate no replies (all known services were already included in the request). Considering this, devices with availability of 9500 s answer almost 50% of the service requests. If other service discovery protocol were used, all devices would answer with equal probability, 20%. This means that with PDP, fixed devices with greater availability time and less limitations answer most of the requests. This was one of the objectives of our protocol. (Fig. 4) also shows that devices with very small availability time (in our case, 500 s) answer more requests than devices with middle availability times. This is because these devices are highly mobile, continually change of networks, and in each new environment they arrive, they have to answer requests above their own services, to

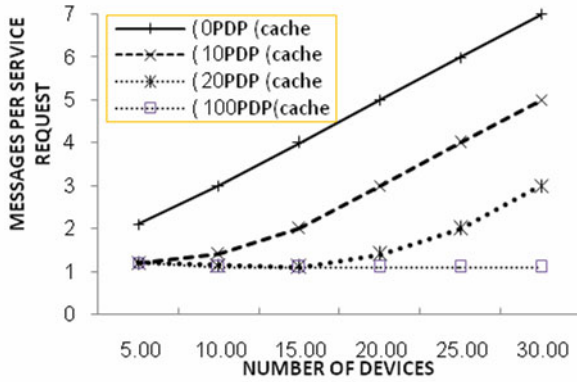


Fig. 3. Service replies per search for different number of devices

make them known to the rest of the devices. As we know, PDP is a fully distributed protocol, it does not rely in any central directory. However, with this simulation we show that PDP is designed time. It is worth mentioning that all percents shown in such a way that, if there are devices that are less mobile (remain more time in the environment) and that have more memory, most of the queries will be answered by them, relieving the more mobile and limited ones of answering, and so preserving their battery. In this scenario we assume that devices with higher availability time also have greater cache's sizes. This is a realistic assumption, since fixed devices use to have more memory than mobile (small, battery powered) devices All the above figures considered PDP one query– multiple responses queries. If the application is interested in the service, not in which device offers it, PDP one query–one response (1/1) can be used instead, obtaining a further reduction in number of messages and power consumption. (Fig. 5) compares PDP one query–one response against the same service discovery protocols as before.

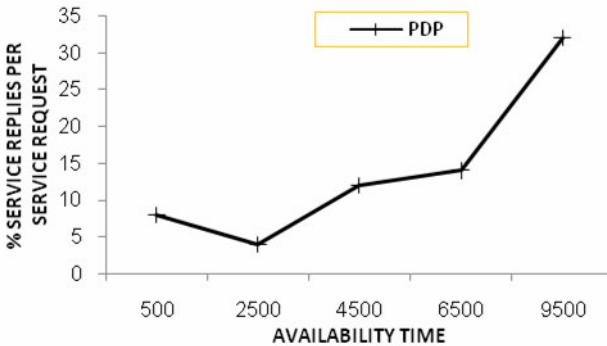


Fig. 4. Service replies per search in an heterogeneous environment with PDP

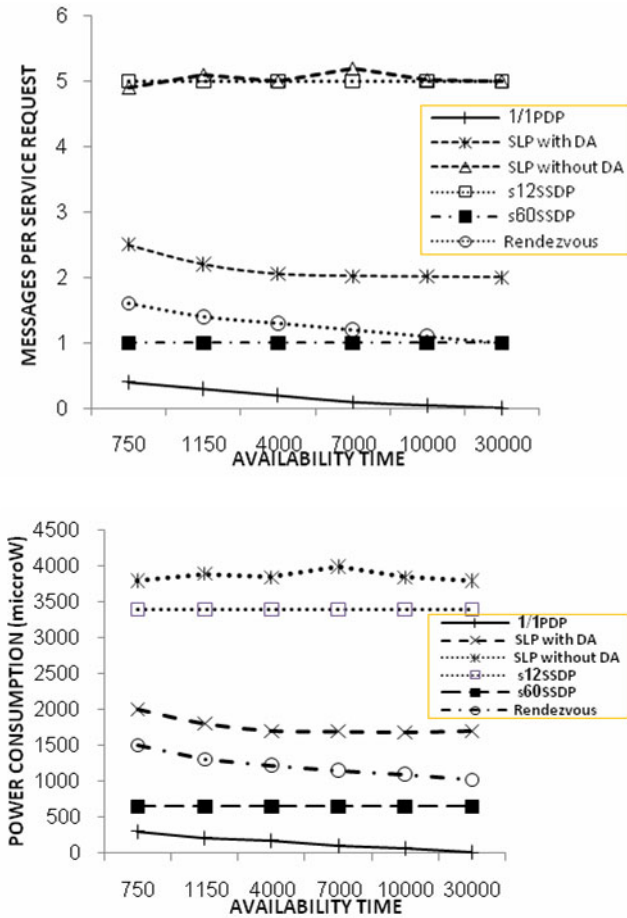


Fig. 5. Comparison of PDP 1/1 with other protocols

5 Conclusions

Ad-hoc networks are becoming increasingly common thanks to the development of mobile device technology. When a device connects to an ad-hoc network, it wants to know the services offered by the network and in turn it may offer its own services. Client applications in the device want to discover trustworthy services automatically, while server applications want to be used by trustworthy clients that will not misuse or attack them. Additionally, secure network communication is also an important issue. These goals are carried out by SPDP. SPDP is a suitable service discovery protocol for ad-hoc networks since:

- It is based on a distributed open architecture; therefore, it does not require central servers;

- It has a simple architecture which contains only two types of components, user agents and service agents;
- It provides autonomous and mobile agents with a simple method for discovering services that are available;
- It minimizes battery use in all devices since the number of transmissions necessary to discover a service is reduced as much as possible;
- It integrates a security model in order to guarantee the security level required by devices. Security issues include authenticity, and data integrity based on a decentralized trust management model.

References

- [1] Guttman, E., Perkins, C., Veizades, J., Day, M.: RFC 2608: Service Location Protocol, Version 2 (June 1999)
- [2] Goland, Y.Y., Cai, T., Leach, P., Gu, Y.: Simple service discovery protocol/1.0. Internet-draft (work in progress), draft-cai-ssdp-v1-03.txt (April 1999)
- [3] Cheshire, S.: DNS-Based Service Discovery. Internetdraft (work in progress) (February 2004)
- [4] Jini Architectural Overview, White Paper (1999)
- [5] Salutation Consortium (1998), <http://www.salutation.org>
- [6] Bluetooth Specification v1.1, Part E: Service Discovery Protocol (SDP)
- [7] Infrared Data Association, Infrared data association link management 1.1 (January 1996)
- [8] Chakraborty, D., Joshi, A., Yesha, Y., Fini, T.: GSD: A novel group-based service discovery protocol for MANETS. In: 4th IEEE Conference on Mobile and Wireless Communications Networks (MWCN 2002), Stockholm, Sweden (September 2002)
- [9] Oh, C.-S., Ko, Y.-B., Kim, J.-H.: A hybrid service discovery for improving robustness in mobile ad hoc networks. In: The International Conference on Dependable Systems and Networks, DSN 2004, Florence, Italy (July 2004)
- [10] Koodli, R., Perkins, C.E.: Service discovery in ondemand ad hoc networks (draft-koodli-manetservicediscovery- 00.txt), Internet-draft (work in progress) (October 2002)
- [11] Engelstad, P.E., Zheng, Y.: Evaluation of service discovery architectures for mobile ad hoc networks. In: 2nd Annual Conference on Wireless On-demand Network Systems and Services (WONS 2005), pp. 2–15 (January 2005)
- [12] Nidd, M.: Service discovery in DEAPspace. IEEE Personal Communications (August 2001)
- [13] Helal, S., Desai, N., Verma, V.: Konark—A service discovery and delivery protocol for ad-hoc networks. In: Third IEEE Conference on Wireless Communication Networks (WCNC), New Orleans (March 2003)
- [14] Barbeau, M., Kranakis, E.: Modeling and performance analysis of service discovery strategies in ad hoc networks. In: International Conference on Wireless Networks, ICWN 2003, Nevada, Canada (June 2003)
- [15] Apple, Rendezvous (2004), <http://developer.apple.com/macosx/rendezvous/>
- [16] Czerwinski, S.E., Zhao, B.Y., Hodes, T.D., Joseph, A.D., Katz, R.H.: An architecture for a secure service discovery service. In: Proc. Mobicom 1999 (1999)
- [17] Helal, S.: Standards for service discovery and delivery. IEEE Pervasive Computing, 95–100 (July/September 2002)

- [18] Toh, C.-K.: *Ad Hoc Mobile Wireless Networks*. In: *Protocols and Systems*, Prentice-Hall PTR, Englewood Cliffs (2002)
- [19] Feeney, L.M., Nilsson, M.: Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In: *IEEE INFOCOM* (2001)
- [20] Morais Cordeiro, C., Gossain, H., Agrawal, D.P.: Multicast over wireless mobile ad hoc networks: present and future directions. *IEEE Network* 17(1), 52–59 (2003)
- [21] Pascoe, B.: *Salutation-lite*. Find-and-bind technologies for mobile devices, Technical Report, Salutation Consortium (June 1999)
- [22] Kaminsky, A.: *JiniME: Jini connection technology for mobile devices*. Technical Report, Information Technology Laboratory, Rochester Institute of Technology (August 2000)
- [23] Li, G.: *JXTA: A network programming environment*, *IEEE Internet Computing*, 88–95 (May–June 2001)
- [24] Dobrev, P., Famolari, D., Kurzke, C., Miller, B.A.: Device and service discovery in home networks with OSGi. *IEEE Communications Magazine*, 86–92 (August 2002)
- [25] Grimm, R., Davis, J., Lemar, E., Macbeth, A., Swanson, S., Anderson, T., Bershad, B., Borriello, G., Gribble, S., Wetherall, D.: *Programming for pervasive computing environments*, Technical Report, University of Washington (June 2001)
- [26] Adjie-Winoto, W., Schwartz, E., Balakrishnan, H., Lilley, J.: The design and implementation of an intentional naming system. In: *17th ACM Symposium on Operating Systems Principles (SOSP1999)*, pp. 186–201 (December 1999)
- [27] Cheshire, S.: *Performing DNS queries via IP Multicast*, Internet-draft (work in progress) (February 2004)
- [28] Esibov, L., Adoba, B., Thaler, D.: *Linklocal Multicast Name Resolution (LLMNR)*, Internet-draft (work in progress) (July 2004)
- [29] Helal, S., Desai, N., Verma, V., Arslan, B.: *Konark: A system and protocols for device independent, peer-to-peer discovery and delivery of mobile services*. *IEEE Transactions on Systems, Man, and Cybernetics* 33(6), 682–696 (2003)
- [30] Campo, C.: *Service discovery in pervasive multi-agent systems*. In: Finin, T., Maamar, Z. (eds.) *AAMAS Workshop on Ubiquitous Agents on Embedded, Wearable, and Mobile Agents*, Bologna, Italy (July 2002)
- [31] Campo, C., Munõz, M., Perea, J.C., Mariñ, A., García- Rubio, C.: *GSDL and PDP: a new service discovery middleware to support spontaneous interactions in pervasive systems*. In: *Middleware Support for Pervasive Computing (PerWare 2005) at the 3rd Conference on Pervasive Computing, PerCom 2005* (March 2005)
- [32] Siva Ram Murthy, C., Manoj, B.S.: *Ad Hoc Wireless Networks: Architectures and Protocols*. Prentice-Hall PTR, Englewood Cliffs (2004)
- [33] Meyer, D.: *RFC 2365: Administratively Scoped IP Multicast* (July 1998)
- [34] Campo, C., Perea, J.C.: *Implementation of pervasive discovery protocol* (2004), <http://www.it.uc3m.es/celeste/pdp/>
- [35] Guttman, E., Perkins, C., Kempf, J.: *RFC 2609: Service Templates and Service: Schemes* (June 1999)

Certificate Path Verification in Peer-to-Peer Public Key Infrastructures by Constructing DFS Spanning Tree

Balachandra¹, Ajay Rao¹, and K.V. Prema²

¹ Dept. of Information and Communication Technology, Manipal Institute of Technology,
Manipal University, Manipal, India

² Dept. Of Computer Science and Engineering, Manipal Institute of Technology,
Manipal University, Manipal, India

bala_muniyal@yahoo.com, ajay041@gmail.com, drprema.mits@gmail.com

Abstract. Authentication of users in an automated business transaction is commonly realized by means of a Public Key Infrastructure(PKI). A PKI is a framework on which the security services are built. Each user or end entity is given a digitally signed data structure called digital certificate. Peer-to-Peer(also called Mesh PKI) architecture is one of the most popular PKI trust models that is widely used in automated business transactions, but certificate path verification is very complex since there are multiple paths between users and the certification path is bidirectional. In this paper, we propose a novel method to convert a peer-to-peer PKI to a Depth First Search(DFS) spanning tree to simplify the certificate path verification by avoiding multiple paths between users, since the DFS spanning tree equivalent of peer-to-peer PKI contains only one path between any two Certification Authorities.

Keywords: Public Key Infrastructure (PKI), Certificate Authority (CA), Cross Certificate, Mesh PKI, DFS Spanning Tree.

1 Introduction

Public Key Infrastructure (PKI), which is also called a *trust hierarchy*, consists of Certificate Authorities, which produce digital certificates and registration authorities, which are used to authenticate and verify the validity of each party involved in an internet transaction. In PKI, the certificates which are issued to the end entities are called end entity certificates. There are some basic extensions that are used to differentiate between certificate authority and end entity certificate [1].

A number of security related services like end entity authentication, data integrity etc. are supported by PKI and this needs a proper utilization of the key pairs generally used in PKI i.e., public or private key pairs. Every certificate should be validated before it is used. Validating a certificate needs the creation of a certification path, or certificate chains, between the certificate and established point of trust. Once the path is created all the certificates which come in this path must be checked. The checking of the certificates which come in the path is referred to as certification path processing [1].

Certificate validation process in general consists of 2 phases, the first one is the construction of the path known as path construction and the other one is validating the

path which is constructed in previous step. The two steps in certificate validation are described in detail below:

- 1) Path Construction- Path Construction is a process which consists of creating or building one or more candidate certificate path. As in [1], the term candidate is used to indicate that even though the certificates are chained together a possibility that the path itself may not be validated due to different reasons as path length, name etc.
- 2) Path Validation- Path Validation is the process in which each certificate in the path is within its established validity period, has not been revoked, has integrity etc. and all the constraints levied on path or all of the certification paths are honored.

A public key infrastructure serves many purposes like enabling the entities to verify public key bindings, binding public keys to the entities and providing a number of services for the management of keys in a distributed system. The protection and distribution of information is one of the main goals of the security architecture which is a must in widely distributed environments, where the users, resources etc. can do their business together even if they are in different places at the same time.

One of the main advantages of using PKI is that it allows to address the security needs and the user will be able to make use of the scalable and distributed characteristics. It also allows the user to be in their e-business with a confidence that the integrity of data has not been compromised and the identity of the receiver and the sender are real [2].

Let us consider a basic scenario where two users staying in different places wish to start a business transaction. Here it might not be necessary that these two users have actually met in person. Since these two users have to achieve security goals, they decide to use public key cryptography. Then they must obtain the public keys of the other user and should be able to authenticate each other. This is usually performed through a trusted third party.

The work of the trusted third party is the distribution of the public keys and authentication. The public key infrastructure consists of a number of different software, encryption technologies, and services, whose main goal is the protection of the security of the user's communication and business transactions. Every user and server in a PKI system will possess a digital certificate, an end-user in enrolment software etc. The PKI is based on public key cryptography and even the name is derived from the same, which is the major technology behind digital signature techniques.

1.1 PKI Components

PKI consists of number of components like certification authorities, registration authorities, repositories and archives. The uses of the PKI can be divided into different categories such as certificate holders and relying parties. Sometimes there can be an optional component called an attribute authority.

- Certification authority (CA): The job of a CA is the identification of the users who are sending and receiving electronic payments, or having communication etc. We can say that authentication is a must in case of a formal communication which includes any kind of cash transaction, like a pin

number in an ATM. This is one of the basic building blocks of the PKI. A number of computer hardware, software and the operators will together form the CA. There are two attributes for a CA [3]; one is its name and the other, public key. The PKI functions of the CA are as given below. It starts with the issue of the certificate, preceded by maintenance of the status information and issues CRLs followed by publishing of the current certificates and CRL's and finally the maintenance of the archives of the status information about the expired certificates. Sometimes the CA may have to distribute these functions to the other components of the infrastructure in order to accomplish the four steps simultaneously.

- **Registration authority (RA):** Once authentication is done by a CA it will ask the RA to register or vouch for the identity of users to a CA. The certificate contents are made in such a way that it will reflect the information presented by the requesting entity and sometimes they also reflect third party information. CA and RA are similar in a way that both contain computer hardware, software and an operator. But a small difference we can say that CA will be mostly operated by multi-user where as RA will be often operated by a single user. Each CA contains a list of its trustworthy RAs. CA identifies RA by a name and a public key. RAs signature on a message means that a CA which has a trustworthy relation with that RA can trust the message. So the RA should be providing an adequate protection for its own private key.
- **PKI Repository:** It is a database for a CA where the digital certificates have been stored. When the users want to confirm the status of the digital certificate for any of the other reason they will contact the repository and the repository will in turn produces digitally signed messages and will send back to the user.
- **Archive:** Archive is a database which will be used with PKI for long term storage of archival information on behalf of CA. It is responsible for giving assurance that the information received was in a good condition and it has not been modified while it was in the archive. CA provides all the information necessary for archive to determine that the certificate is original that is issued by the same CA specified in the certificate and it is valid.

A user doesn't issue certificate, instead they rely on the other components of the PKI to obtain the certificates. A user can be a group or an individual who make use of the PKI. Even for the verification of the certificates user has to depend on the other components of the PKI. A user, whether it be an individual or a group they hold certificates for various applications.

1.2 PKI Models

There are many types of PKI architectures. Few popular models are described below:

1. **Hierarchical Model:** Hierarchical model uses the basic tree structure. There will be a CA which controls and provides information to all the other CA's, which is called a Root. In this model the root CA is called a "trust anchor" [4]. Here the root CA can have one or more children CAs and each child will be having their own children CAs. But the leaf node will be the user which

requests the certificate from the CAs. The certification path is as follows: the root issues certificate to its children and each CAs in the level one issues certificate to other CAs or users in the next level. Advantages of this hierarchical PKI model are, it is scalable and Certificate path construction is simple.

2. Mesh PKI: It is a peer to peer PKI model and it is also called web of trust. Here each subscriber trusts the CA that issued their certificates. They use a special type of certificate which contains the information about the security policy transformation, which is involved in moving from one domain of trust to another. Since there are a multiple paths between the users, it is more complex to construct a certificate path. One advantage is that the compromise of a single CA cannot bring down the entire PKI.
3. Bridge PKI: This is a different approach for the interconnection of the PKIs. Here they use a special authority called Bridge Certificate Authority (BCA). This model is used in order to reduce the number of the cross certification paths found in the mesh model. In this, there will be a central cross certification authority which will be connected to all the other CA's.
4. Hybrid PKI: As its name implies the hybrid PKI is interconnection of different PKIs through cross certification. There are three types of cross certifications distinguished by an ISO Hybrid Model. The first one is hierarchical cross certificate [5] which parallels the paths of the subordinated hierarchies but extend upward towards the root CA's. Another one is general cross-certificate [5] which interconnects the CAs either at the root level or between the points within the connected hierarchies. The last one is the special cross certificate[5] which is intended to allow selective establishment of certification paths that may not conform to the restrictions ordinarily imposed hierarchically along the path from the root CA's.
5. Trust Model Based Gateway CA's: The certification to different kinds of CAs which is located in different places can be provided by only a gateway CA. Each of the GWCA's has their own trust regions. All the GWCA's are connected in a ring fashion. When the GWCA is connected in a ring configuration its subordinate CA's can be connected using different models such as mesh, bridge etc. Here in this model GWCA are the most trusted anchors to their respective end entities.

1.3 Cross Certification

The Cross Certification between CAs [6] can be performed in two basic steps:

- The first step deals with the establishment of a trust relationship between two CAs through the signing of another CA's public key in a certificate. This particular certificate is referred to as cross certificate. This is generally executed in different time intervals.
- The second step deals with the verification of the trust worthiness of a user certificate which is signed by a CA. This CA should be in the same PKI network. This step is generally referred to as "walking a chain of trust" [6]. The term chain of trust indicates a number of cross validations which are either of the roots CA or the trust anchor [6] of the verifying user required to validate the other user certificate.

As referred in [6] a trust anchor is generally a key which is used for the verification of the CA by the client application as the validation. The difference of the hierarchical cross certification to peer to peer cross certification is the location of the users trust anchor vis-à-vis the user.

- The user’s local CA is a subordinate CA in hierarchy of CAs, if the users trust anchor is not the user’s local CA. The public key for root CA is the user’s trust anchor. The subordinate CAs can add more subordinate CAs to the hierarchy below it, even if it cannot perform the cross certification with other CAs, if permitted by policy. All certificate validation in the hierarchy starts with the root CA’s public key. The Fig1 shows basic hierarchical cross certification model.

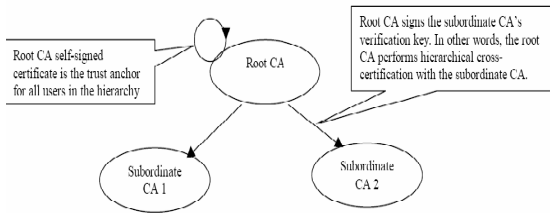


Fig. 1. Hierarchical Cross-Certification Between a root (autonomous) CA and subordinate (non-autonomous) CAs

- The user’s local CA is an autonomous CA if the user’s trust anchor is the user’s local CA. Autonomy refers to the fact that the CA does not rely on a superior CA in a hierarchy. Here any CA can perform cross certification with other CA and the CA is superior to himself. Here there is no concept of subordinate CAs. All certificate validation for clients within an autonomous CA starts with the local CA’s self-signed certificate.

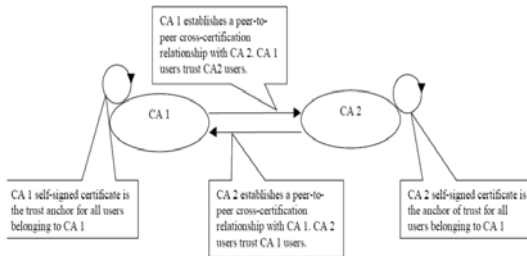


Fig. 2. Peer to Peer Cross Certification between two CAs

2 Related Work

In mobile ad hoc networks, the CAs are considered to be dynamic in nature, and the central PKI service i.e. the root CA may be unavailable for some time. But the assistance

from intermediate CAs is needed for successful communication between the requestor and the requested node. Any request message for communication by any node or user should pass through the root CA. So there will be bottleneck in the root CA and excessive usage of bandwidth causes serious problems in the network. When the request message is sent and the root CA is not available for some reason, much of the bandwidth used to send that particular message is wasted and that in turn reduces the efficiency of the model. This is the drawback of the hierarchical model in wireless MANET.

Christina et al. [7] propose a scheme of building a virtual hierarchy to simplify the certification path discovery in mobile ad hoc network. They developed an algorithm called PROSEARCH for this purpose. In peer to peer model, the path discovery is difficult because the trust relationship is bidirectional, where as in hierarchical, it is unidirectional. So the algorithm converts a peer to peer model to a virtual hierarchical model. By doing this, path discovery is simplified and maximum path length can be adapted to the characteristics of the user within the limited storage and processing capacity.

3 Path Verification in Mesh PKI

In a peer-to-peer model or a Mesh architecture, each CA is autonomous, because the CA does not rely on a superior CA as in the case of hierarchy. The user's trust anchor is its local CA and all the CAs can be the trust points. Here any CA can perform cross-certification with other CAs and the CA is superior for himself. Here there is no concept of subordinate CAs. All certificate validation for clients within an autonomous CA starts with the local CA's self-signed certificate.

Sometimes, a certificate may contain additional information about the degree of trust. For example, if a CA wishes to limit its trust, it must specify these limitations in the certificates issued to its peers. In a Peer-to-Peer PKI, the certificate path construction starts from the local CA's self-signed certificate. One of the advantages of a Peer-to-Peer PKI is that, there is no single point of failure and can easily incorporate a new community of users. But, as shown in Fig3, there can have multiple paths between two users resulting in certificate path verification complex and ambiguous. In addition, a peer-to-peer PKI can easily be constructed from a set of isolated CAs because users do not need to change their trust points. If there is dynamic changes in structures or environments, the organizations can definitely go for deploying this PKI model because in this case the communicating entities are not related hierarchically. If the number of CAs is n and the graph representing Peer-to-Peer PKIs is complete, then the number of trust relationships is directly proportional to the number of CAs, that is, the number of trust relationships is equal to $n*(n - 1)$, that causes scalability problems. In addition, this model requires larger certificates because the users must determine which applications a certificate may be used for based on the content of the certificates. All the above facts about Peer-to-Peer PKIs make the certificates with more extensions and so the certificate path verification and validation process is more complex. In the worst case, the maximum length of a certification path in a peer-to-peer PKI is the number of CAs in the infrastructure.

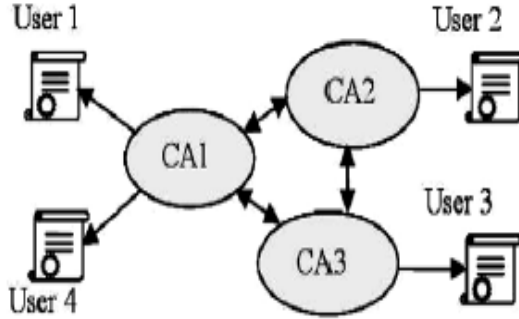


Fig. 3. Peer-to-Peer network

4 Proposed Method of Converting Peer-to-Peer PKI to DFS Spanning Tree to Simplify Certificate Path Validation

In automated business transactions, the method of building a trusted path between the trusted anchor and the target entity constructs a path as each certificate is retrieved from a repository via LDAP, a protocol employed for repository access operations. The simplicity of such method resides in the fact that only one certification path is possible in the case of Hierarchical PKI. Also the path is unidirectional and simple.

But, building a path in Peer-to-Peer PKI is nothing but traversing a complex graph. However, from the simplest viewpoint, writing a path-building module can be nothing more than traversal of a spanning tree, even in a very complex cross-certified environment. In the proposed method, we traverse the graph representing Peer-to-Peer PKI in Depth First Search (DFS) [8] order and construct a DFS spanning tree. In a DFS spanning tree, we have a single path between any two users. We construct DFS spanning tree as follows:

Let graph G represent the Peer-to-Peer PKI in question. Each vertex in the graph represents a CA. We shall start from a given vertex v in the graph, G . We will mark this vertex v as visited. The next step is to pick a new unvisited vertex (any one of the adjacent vertices can be selected). Call this vertex w . We then explore this new vertex depending upon its adjacent vertices recursively. The search procedure terminates after all the vertices are explored (i.e. visited). As each vertex is visited, it is added to the DFS spanning tree.

Algorithm Construct_DFSSpanningTree(v)

```

{
  //G is a Graph representing Peer-to-Peer PKI. Each edge is
  bidirectional
  //Visited[1:n] is an array to remember the visited information
  // v is the starting vertex

  Visited[v]=1; //mark the starting vertex visited
  Add v to the DFS spanning tree
  for(each vertex w adjacent to v)

```

```

    if(visited[w]=0)
        Construct_DFSSpanningTree(w); //continue to explore
    }

```

For example, Figure 4(a) is a graph that represents a Peer-to-Peer PKI in which the path is bidirectional. The DFS order of the tree is: CA1, CA2, CA3, CA6, CA4, CA5, and CA7.

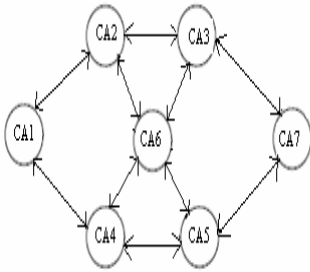


Fig. 4(a). A Peer-to-Peer PKI

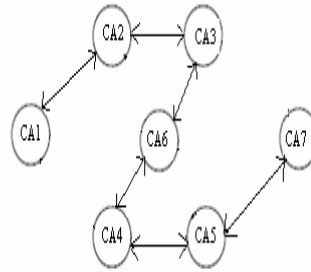


Fig. 4(b). DFS spanning tree

Fig 4(b) represents the DFS spanning tree of the graph shown in Fig 4(a). From Fig 4(b), we can observe that there exists single path between any two CAs. Since there exists single path between any two CAs in the spanning tree, the certificate path construction is simple and straight forward. The complexity due to multiple paths between CAs and ambiguity in choosing one of them is removed.

5 Simulation Results

The algorithm is implemented in Java platform using NetBeans IDE on a Windows Vista/XP machine with 2 GB RAM. Peer-to-Peer PKI performance evaluation is performed based on the time elapsed on path validation between different users under different nodes.

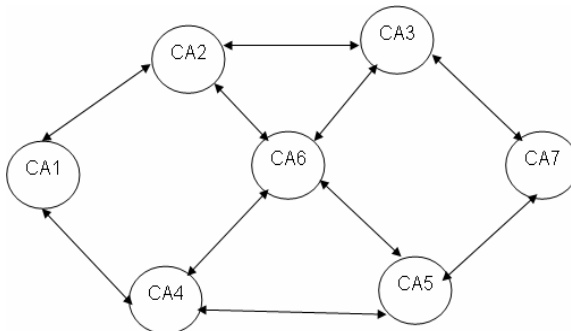


Fig. 5. A Peer-to-Peer PKI

One of the special features of Mesh PKI is that every node may be connected to every other node i.e. if there are n number of nodes, then the links needed are $n*(n-1)$. Here the time required for path validation does not depend upon the number of nodes. We considered the Mesh PKI shown in Fig5 as the input for the implementation.

The graph shown in Fig6 illustrates the certificate verification time required by a user certified by CA1 to verify the certificates of the users certified by different CAs and the time required for the same in the equivalent DFS spanning tree.

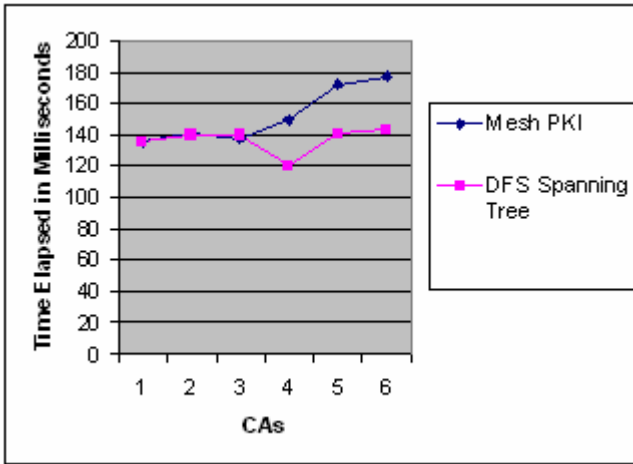


Fig. 6. Comparative Analysis of Path verification time of Mesh PKI and its Equivalent DFS Spanning Tree

6 Conclusion

In mesh or Peer-to-Peer PKI, certificate path verification is a complex task since there exists multiple paths between CAs. In this paper, we proposed an efficient method to convert a mesh or Peer-to-Peer PKI to its equivalent DFS spanning tree to simplify the certificate path construction. Using this algorithm, the complexity of certificate path verification in Peer-to-Peer PKIs due to multiple paths between users can be removed. The main benefit of this approach is that the needed certificate for a complete certificate path can be computed in advance, thus accelerating certificate collection activity.

References

1. Lloyd, S.: Understanding certificate path construction. PKI Forum (September 2002)
2. Richard Kuhn, D., Hu, V.C.: Introduction to Public Key Technology and the Federal PKI Infrastructure, National Institute of Standard and Technology, U.S. Government publication (February 26, 2007)
3. Hesse, P.M., Lemire, D.P.: Managing interoperability in non-hierarchical public key infrastructures. Gemini Security Solutions

4. Guo, Z., Okuyama, T., Finley Jr, M.R.: A New trust Model for PKI Interoperability, Information Network Laboratory, Graduate School of Business Administration Asahi University, Japan
5. Linn, J.: Trust models and managements in Public Key Infrastructures, RSA Laboratories, 20 Crosby Drive Bedford, MA 01730 USA, jlinn@rsasecurity.com, (November 6, 2000)
6. Turnbull, J.: Cross-certification and PKI policy Networking, Version 1.0, Entrust Securing Digital Identities And Information (August 2000)
7. Satizabal, C., Hernandez-Serrano, J., Forne, J., Pegueroles, J.: Building a virtual hierarchy to simplify certification path discovery in mobile ad hoc network, Department of Telematics Engineering, Technical University of Catalonia, Jordi Girona 1-3 C3, 08034 Barcelona, Spain (January 9, 2007), <http://www.elsevier.com/locate/comcom>
8. Sahni, S.: Data Structures, Algorithms and Applications in C++", Magraw Hill International edition (2000)

A Novel Deadlock-Free Shortest-Path Dimension Order Routing Algorithm for Mesh-of-Tree Based Network-on-Chip Architecture*

Kanchan Manna¹, Santanu Chattopadhyay², and Indranil Sen Gupta¹

¹ School of Information Technology

² Dept. of Electronics and Elec. Comm. Engg., Indian Institute of Technology
Kharagpur, India - 721 302

{kanchanm@sit,santanu@ece, isg@cse}.iitkgp.ernet.in

Abstract. This paper presents a new dimension ordered routing algorithm for Mesh-of-Tree (MoT) based Network-on-Chip (NoC) designs. The algorithm has been theoretically proved to be deadlock, live-lock and starvation free. It also ensures shortest-path routing for the packets. The simplified algorithm, compared to the previously published works, provides same throughput and average latency measures, at a lesser hardware overhead (about 61% for routers, 46% for links, and 44% in total) due to possible reduction in the minimum flit-size. It allows us to vary router complexity more flexibly while planning the MoT based NoC for application specific System-on-Chip (SoC) synthesis.

Keywords: MoT, NoC, Interconnection Architecture, Routing.

1 Introduction

Network-on-Chip (NoC) has evolved as a viable alternative for on-chip communication. In this, a pre-designed network-fabric consisting of routers and links connecting them in a definite topology is utilized. The intellectual property (IP) cores are connected to the routers. The cores exchange messages (instead of wires) between themselves through this on-chip network fabric. Network interface (NI) module is put as interface between an IP core and the corresponding router.

Designing a routing algorithm is the most crucial part in NoC architecture design. A good routing algorithm should provide three features: low communication latency, high network throughput, and ease of implementation in hardware [5]. In this paper we have proposed a novel Dimension Order Routing (DOR) algorithm that enables us to reduce latency, power, and area requirements for the NoC [1]. DOR is based on ordering the dimensions in a network and routes the packets strictly in the same order [5]. Applied to two-dimensional Mesh-of-Tree (MoT) [3], it produces *xy routing algorithm*; route a packet first along the x dimension

* This work is partially supported by Department of Science and Technology, Govt. of India (SR/S3/EECE/0012/2009), Dt 20th May, 2009.

(row) and then along the y dimension (column). We also show that the algorithm is free from *Deadlock*, *Starvation*, and *Live-lock* [5]. The algorithm ensures simple addressing scheme, simpler routing algorithm, and reduced area overhead compared to [2, 3, 4]. The paper organized as follows. Section 2 describes earlier works and highlights our contribution. In Section 3, we have discussed about addressing scheme. The new algorithm is presented in Section 4. In Section 5, we have discussed about guaranteed shortest path. Section 6 proves that the algorithm is free from deadlock, starvation, and live-lock. In Section 7 we have discussed about metrics for performance evaluation. Simulation methodology has been discussed in Section 8. Simulation result is described in Section 9. Finally in Section 10, we conclude the work.

2 Related Works and Our Contributions

A good NoC architecture should have the following three features: large bisection width (for parallel communication), small diameter (for reduced communication delay), and lower node degree (lesser metal layers) [3]. There are several kinds of interconnection topologies reported in the literature. They include Cliche (Mesh), Torus, Folded Torus, Binary Tree, Octagon, Scalable Programmable Integrated Network (SPIN), Butterfly Fat Tree (BFT) and Mesh-of-Tree (MoT) [6] - each has its own advantages and disadvantages [2]. A hybrid of mesh and tree topologies is Mesh-of-Tree (MoT). It was first proposed by Balkan et al [6] for NoC architecture to communicate from processors (sources) to memory modules (destinations). The trees are categorized as row tree or column tree. They place the processors and memories at the root of row trees and column trees respectively. It thus requires a large number of routers.

Kundu and Chattopadhyay [2, 3, 4] proposed a modified MoT configuration (Fig. 1(b)). In their configuration MoT has three types of routers – root, stem, and leaf. IP cores are connected to leaf routers only. In this connection, a 4x4 configuration with 32 cores, need a total of 40 routers, out which 16 are leaf routers, 16 stem routers, and remaining 8 are root routers. Each leaf has 2 cores. For a 4x4 MoT, a total 13 bits have been utilized for identifying a source or destination core. This has forced a minimum flit length to be 32 bits - 13 bits for destination, 13 bits for source, 1 bit each for end of packet (EOP) and begin of packet (BOP), 2 bits for virtual channel (VC), 1 bit for message type (e.g. request-response type of message) and 1 bit for identifying quality-of-service priority. Naturally the buffers within routers are 32 bit wide. Our work presents an improved routing scheme that reduces the number of address bits needed in MoT significantly (from 13 to 5 for a 4x4 MoT). This plays a major role in reducing the network complexity. The minimum flit size can now to reduced to 16 bits which will definitely have positive impact over area and power profile of the NoC. Novel contributions of this paper are as follows. We present an improved addressing scheme for MoT routers. Next a novel MoT routing algorithm based on dimension order routing has been proposed. Routing algorithm has been shown to be deadlock, starvation, and live-lock free and to follow the shortest

path. The router has been synthesized in Synopsys Design-Vision to get area estimate. It shows significant area reduction, compared to [2, 3, 4]. Performance of a 4x4 network has been evaluated.

3 Addressing Scheme

An example 4x4 MoT is shown in Fig. 1. The leaf routers marked as ‘L’. A leaf level router connects to a stem level row router (marked ‘RS’) and to a stem level column router (marked ‘CS’). The stem level routers are further connected to root level router (marked ‘R’). In our addressing scheme we identify only leaf routers and cores. We allocate an address to a leaf router based on its position in the two dimensional grid (matrix). We allocate a group address to a stem router. The group address is decided by the neighboring routers. For row stem (RS), group address will be decided by common bit (s) of column number of neighboring routers, located one level lower than RS. Similarly, column stem (CS) group addresses are decided. Since each leaf router can have two cores attached to it, complete address for a core would be [core_id,column_number,row_number].

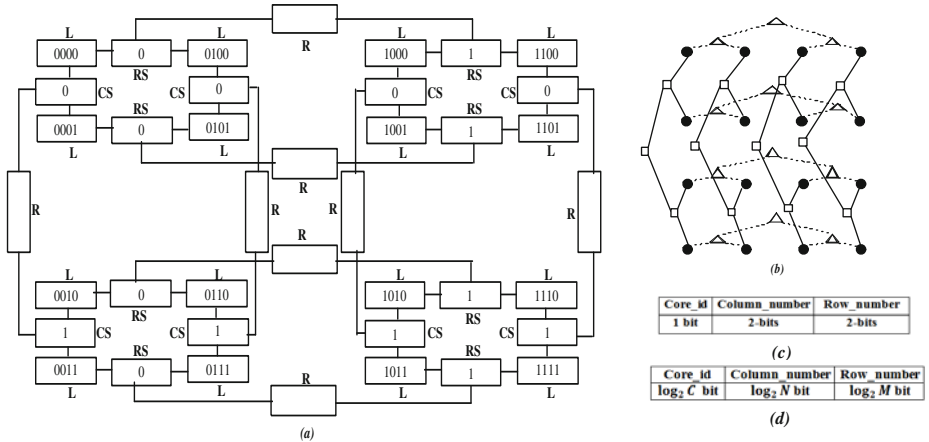


Fig. 1. (a) Proposed addressing scheme for 4x4 MoT (b) 4x4 MoT topology (c) Addressing format for 4x4 MoT (d) Addressing format for MxN MoT

4 Routing Algorithm

We first present the routing algorithm for a leaf level router. The leaf level router has its own address (CURRENT_ADDR in the algorithm). A leaf router is connected two cores (CORE_LINK) and two stem level routers (ROW_LINK and COL_LINK). A packet with destination address DESTINATION_ADDR is routed via the router as follows. The row part of DESTINATION_ADDR is matched with row part of CURRENT_ADDR. The variable ROW_CHECK is set accordingly. Similarly, the variable COL_CHECK is set by comparing the column parts of DESTINATION_ADDR and CURRENT_ADDR. If both ROW_CHECK

and `COL_CHECK` are *true*, the packet has reached the destination router, in which case it is forwarded to appropriate core based on the `CORE_ID` field of `DESTINATION_ADDR`. Otherwise, if `ROW_CHECK` is *true*, the packet is in the row to which it belongs. It is forwarded to the row stem(`ROW_LINK`). Otherwise, the packet is routed to the column stem (`COL_LINK`). Next we present the routing algorithm followed by a stem-level router at level l . We consider the leaf-level routers to form level-0 ($l=0$) of the tree. A stem-level router at level l connects to two ($l-1$)-level routers - we call them `LEFT_LINK` and `RIGHT_LINK` respectively. It also connects to another router at level ($l+1$), we call it `UP_LINK`. As noted earlier, a stem routers possesses a group-id. If a packet with destination address `DESTINATION_ADDR` reaches a stem-router of row tree at level l , the group-id of the packet is given by the most significant l bits of the column number of `DESTINATION_ADDR`. For a column-tree router the value is obtained from the row number part of `DESTINATION_ADDR`. This is accomplished by the function *router_group_addr()* in the algorithm. If the value does not match with the group-id of the router, the packet is forwarded to the `UP_LINK`. Else, the packet belongs to a core in the tree rooted at this stem router. If the $(l-1)^{th}$ bit (counted from MSB) is 0, the packet is routed to the `LEFT_LINK`, else, the packet is routed to the `RIGHT_LINK`.

The root routers do not need any routing algorithm. They just copy the packet arriving on one link to the other, acting like a buffer.

Algorithm 1. Proposed Algorithm for Leaf Router

Input: `DESTINATION_ADDR`

Output: Generate request for `CORE`, or `ROW_LINK`, or `COL_LINK`

if `row(CURRENT_ADDR) = row(DESTINATION_ADDR)` **then**
 `ROW_CHECK` \leftarrow *true*

end

else

`ROW_CHECK` \leftarrow *false*

end

if `col(CURRENT_ADDR) = col(DESTINATION_ADDR)` **then**

`COL_CHECK` \leftarrow *true*

end

else

`COL_CHECK` \leftarrow *false*

end

if `ROW_CHECK = true and COL_CHECK = true` **then**

 Send packet to `CORE_LINK` base on `CORE_ID` of
 `DESTINATION_ADDR`.

end

else if `ROW_CHECK = true` **then**

 Route packet through `ROW_LINK`.

end

else

 Route packet through `COL_LINK`.

end

Algorithm 2. Proposed Algorithm for l^{th} level Stem Router

```

Input: DESTINATION_ADDR
Output: Generate request for UP_LINK, or LEFT_LINK, or RIGHT_LINK
if GROUP_ID = router_group_add(DESTINATION_ADDR) then
  UP_LINK  $\leftarrow$  false
end
else
  UP_LINK  $\leftarrow$  true
end
if UP_LINK = false then
  if extract_bit(DESTINATION_ADDR,  $l - 1$ ) = 0 then
    (*  $l$  indicates at which level of tree the stem is present.*)
    Route packet through LEFT_LINK.
  end
  else
    Route packet through RIGHT_LINK.
  end
end
else
  Route packet through UP_LINK.
end

```

5 Guaranteed Shortest Path

All possible transactions in an $M \times N$ MoT can be enumerated as follows.

1. Source and destination cores are in same leaf router.
2. Source and destination cores are in same row but in different columns.
3. Source and destination cores are in same column but in different rows.
4. Source and destination cores are in different rows and columns.

Here, we will ensure that packet will always reach to the destination using shortest path. We are assuming that there is no error in the router and link. In our addressing scheme we have allocated address to a router based on its position (column number, row number). In a particular row or column, addresses are in strictly increasing order. Thus, for routers at row ' x ' addressing order will be $(y_0, x) < (y_1, x) < (y_2, x) < (y_3, x) < \dots < (y_N, x)$, y_i identifying the column number of the i^{th} router. For a particular column, say ' y ', addressing order will be $(y, x_0) < (y, x_1) < (y, x_2) < (y, x_3) < \dots < (y, x_M)$. In a dimension order routing, shortest path is ensured if the traversal does not violate the ordering. According to our algorithm, a $M \times N$ MoT path between any source and destination will be formed based on one of the comparison sequences noted next.

1. $[\{(Y_r = C_r) \wedge (Y_c = C_c)\} \rightarrow PC]$
2. $[\{(Y_r = C_r) \wedge (Y_c \neq C_c)\} \rightarrow RL] \vdash [\{Y_c(m : 1) > G(RS_1)\} \rightarrow UP_1^+] \vdash [\{Y_c(m : 2) > G(RS_2)\} \rightarrow UP_2^+] \vdash \dots \vdash [\{Y_c(m : l) > G(RS_l)\} \rightarrow UP_l^+] \vdash [Y_c(l) \rightarrow (L_0|R_1)] \vdash [Y_c(l-1) \rightarrow (L_0|R_1)] \vdash \dots \vdash [Y_c(0) \rightarrow (L_0|R_1)] \vdash [Apply\ sequence\ 1]$

3. $[(Y_r \neq C_r) \rightarrow CL] \vdash [\{Y_r(m : 1) > G(CS_1)\} \rightarrow UP_1^+] \vdash [\{Y_r(m : 2) > G(CS_2)\} \rightarrow UP_2^+] \vdash \dots \vdash [\{Y_r(m : l) > G(CS_l)\} \rightarrow UP_l^+] \vdash [Y_r(l) \rightarrow (L_0|R_1)] \vdash [Y_r(l-1) \rightarrow (L_0|R_1)] \vdash \dots \vdash [Y_r(0) \rightarrow (L_0|R_1)] \vdash [Apply\ sequence\ 2]$
4. $[\{(Y_r = C_r) \wedge (Y_c \neq C_c)\} \rightarrow RL] \vdash [\{Y_c(m : 1) < G(RS_1)\} \rightarrow UP_1^-] \vdash [\{Y_c(m : 2) < G(RS_2)\} \rightarrow UP_2^-] \vdash \dots \vdash [\{Y_c(m : k) < G(RS_k)\} \rightarrow UP_k^-] \vdash [Y_c(k) \rightarrow (L_0|R_1)] \vdash [Y_c(k-1) \rightarrow (L_0|R_1)] \vdash \dots \vdash [Y_c(0) \rightarrow (L_0|R_1)] \vdash [Apply\ sequence\ 1]$
5. $[(Y_r \neq C_r) \rightarrow CL] \vdash [\{Y_r(m : 1) < G(CS_1)\} \rightarrow UP_1^-] \vdash [\{Y_r(m : 2) < G(CS_2)\} \rightarrow UP_2^-] \vdash \dots \vdash [\{Y_r(m : l) < G(CS_l)\} \rightarrow UP_l^-] \vdash [Y_r(l) \rightarrow (L_0|R_1)] \vdash [Y_r(l-1) \rightarrow (L_0|R_1)] \vdash \dots \vdash [Y_r(0) \rightarrow (L_0|R_1)] \vdash [Apply\ sequence\ 4]$

where m : Most Significant Bit (MSB), $0 \leq l \leq \log_2 N$, $0 \leq k \leq \log_2 M$.

Table 1. Symbols used in the sequences

Symbol	Description
Y_r	Row bits of destination address
Y_c	Column bits of destination address
C_r	Row bits of current leaf router's address
C_c	Column bits of current leaf router's address
$G(S_l)$	Would return group address of a stem router (S) at level l
$Y_r(a : b)$	Bits from position a to b (including both) of Y_r
$Y_r(a)$	Bit at position a of Y_r
$L_0 R_1$	Take left; if input is 0, Take right; if input is 1
RL	Row Link
CL	Column Link
PC	Packet Consumption
UP_i^+	Move upwards at level i , when packet moving in increasing order
UP_i^-	Move upwards at level i , when packet moving in decreasing order
\vdash	Next router
$(a b)$	Either a or b

Comparison sequence (1) signifies that the row number and column number of destination address are same as the address of the current leaf router. So packet is for consumption by a core attached to that router. Comparison sequence (2) corresponds to the case in which row number of destination address is same as that of the current leaf router, but column numbers do not match. It implies that the packet has to move towards row link (RL), means packet will be forwarded based on column number. Row stem at level-1 will receive the packet. It will compare group address with destination address. Destination's group address bits will consist of MSB to bit 1 of column bits. The group address of stem router will be given by $G(RS_1)$. If stem's group address is lesser than group address returned by $G(RS_1)$, the packet will be forwarded to up direction (UP_i^+). Here '+' indicates that packet is moving from a lower column number to a higher

column number and i indicates the level to which the packet will move to. In this way packet will reach its corresponding level, where group address of the stem router will match with group address of the destination address. After that, the packet will come down level by level through left or right link up to level-0, as in all these cases group address will match. The packet will move based on the l^{th} bit of column number Y_c , l being the level of the router. At level 0, packet will come to leaf router and then based on condition (1) packet will reach the core. Similar explanation can be given for sequence 3, 4, and 5.

All these comparison sequences maintain an order that is either increasing or decreasing. If it encounters a path with comparison sequence not strictly increasing or decreasing, (i.e. a mix of these two), then that path could not be a shortest path. That comparison sequence implies there is a violation in dimension ordering. But we have already noted that dimension order is proper and assumed the router to be error free. Hence our algorithm always provide shortest path.

6 Freedom from Deadlock, Starvation, and Live-lock

A situation where a packet waits for an event that cannot happen is called deadlock. For example, a packet may wait for a network resource (bandwidth or FIFO) to be released by a packet that is, in turn, waiting for the first packet to release some resource. In worm-hole switching, such a circular wait condition causes deadlock. In dimension ordered routing algorithm, circular wait means violation in dimension ordering. As dimension is already ordered, routing follows shortest path technique. So our algorithm is deadlock free.

Starvation is similar to deadlock. This is a situation when a packet waits for an event that can happen but never does. For example, a packet may wait forever to get a network resource. Some other packets are also competing for the same network resource, but those packets always successfully get that resource. Starvation is possible only when the distribution of network resources is not uniform. Packets cannot be injected into the network or move inside network because of deadlock and starvation. We are using Round-Robin arbitration technique, which allocates the channel uniformly. So our routing technique is starvation free.

In contrast, live-lock does not stop the movement of a packet in network, but rather stops its progress toward the destination. Live-lock happens, when routing is adaptive and non-minimal [5]. Ordering the dimensions not only ensures that the xy routing is free from deadlock, but also its non-adaptiveness. So our algorithm is free from live-lock as well.

7 Performance Metrics

For evaluating a NoC architecture, we will have followings metrics [7]: Throughput, Latency, Area Overhead, and Energy Consumption. Throughput is a measure of how much data can be transmitted across a network. For a message passing system throughput T is given by, $T = (Total\ messages\ completed \times$

$Message\ length)/(Number\ of\ IP\ cores \times Time)$. Throughput is measured in terms of flits/cycle/IP. For our case 32 number of IP cores are used and message length is 64 flits.

Network bandwidth refers to the maximum number of bits that can be sent successfully to the destination through the network per unit time. It is represented as bits/sec (bps). It defines as follows. $BW = ((Throughput \times No.\ of\ IP\ cores \times flits\ size))/(Clock\ period)$. Latency can be defined as the time interval (in terms of clock cycle) between transfer of header flits by the source and the receipt of tailer flit by the destination. Let P be the number of packets received in a given time period and let L_i be the latency of the i^{th} packet. Average latency is defined as, $L_{avg} = (\sum_{i=1}^P L_i)/(P)$. In NoC design, area overhead comes due to routers, repeaters, network interface, and links. To estimate the silicon area occupied by each router, we have developed their Verilog models and synthesized using *Synopsys Design Vision in 90 nm* CMOS technology supporting *Faraday* library to generate gate-level net-list. The synthesis tool reports the silicon area occupied by the design in μm^2 .

8 Simulation Methodology

For evaluating the performance of the interconnection network, we have designed a cycle-accurate System-C based simulator. Cores are modeled via traffic generators and receptors. Traffic generator uses self-similar model for generating the traffic. Real traffic follows bursty nature. Self-similarity is also bursty in nature. We categorized traffic as uniform and locality based. For example, if locality factor is 0, then traffic from a core will distributed uniformly to all other core. If locality factor is 0.3 then 30% traffic will go to nearer cores and remaining 70% will go to other cores. For 4x4 MoT, the possible distances (d) of the destinations from any source are $d = 0, 2, 4, 6,$ and 8 . There is only one destination core at the nearest cluster. We derived traffic for our simulation from the traffic generated by model noted in [4]. Each packet contains 64 flits.

9 Experimental Results and Analysis

In this section, we have applied the above mentioned evaluation methodology to find out the performance and area overhead of MoT based network consisting of 32 cores by varying offered loads (N) in self-similar traffic.

9.1 Accepted Traffic vs. Offered Load

The accepted traffic depends on the rate at which the IP blocks inject traffic into the network. Ideally, accepted traffic should increase linearly with increasing load at a slope of 45° . However, due to the limitation of routing and arbitration strategy and unavailability of enough buffer space within the wormhole router, the network suffers from contention. Therefore, the accepted traffic saturates after a certain value of the offered load. Fig. 4(a) depicts this scenario for uniformly

distributed traffic in MoT based network. The maximum accepted traffic where the network is saturated is termed as throughput as defined above and it relates to the maximum sustainable data rate by the network. In Fig. 4(a) and others, the graphs marked “16 bits flit” are our results, where as, the graphs marked “32 bits flit” are those corresponding to [2, 3, 4]. Science throughput is not dependent on flit size the value do not differ.

9.2 Throughput vs. Locality Factor

The effect of traffic spatial localization on network throughput is shown in Fig. 4(b) and 4(c). As the locality factor increases, more number of traffic are destined to their local clusters, thus traversing lesser number of hops which in turn increases the throughput.

9.3 Average Overall Latency vs. Locality Factor

The average overall latency of any network depends on both the offered load and the locality factor. Fig. 5(a) shows the average overall latency profile for uniformly distributed offered load in MoT. It shows that at lower traffic, the latency variation is not significant. This is due to the fact that at lower traffic, the contention in the network is less, but it will increase as the offered load increases, which in turn increases the latency. The simulation result shows that as the offered load increases towards the network saturation point, the latency increases exponentially which signifies that the packets will take much longer time to reach their destinations. So, it is always desirable to operate the network below its saturation point. The effect on traffic spatial localization on average overall latency is also studied and is shown in Fig. 5(b) and 5(c). As the locality factor increases, more number of traffic will go to their local clusters, hence lesser contention in the network. Therefore, the network will be able to carry more traffic before going to saturation, which in turn enhances the operating point of the network.

9.4 Bandwidth Variation

Fig. 6 shows the bandwidth variation of our approach and compares with [2, 3, 4]. Since our flit size is 16-bit, compared to 32-bit for the previous work, bandwidth is also halved. However it should be noted that our router design does not put any restriction on flit size, which can be increased at will, and thus improving the bandwidth. The previous work [2, 3, 4] cannot have a flit size lesser than 32 bits due to address size restrictions.

9.5 Area Overhead

With a restriction in flit size, the FIFO width of the routers also decreases in our work compared to [2, 3, 4]. The simplicity of routing also adds to this reduction. Fig. 7 shows the area reduction. We are getting 61% area reduction in router design and 46.38% area reduction in link area over [2, 3, 4]. The overall area improvement is about 45%.

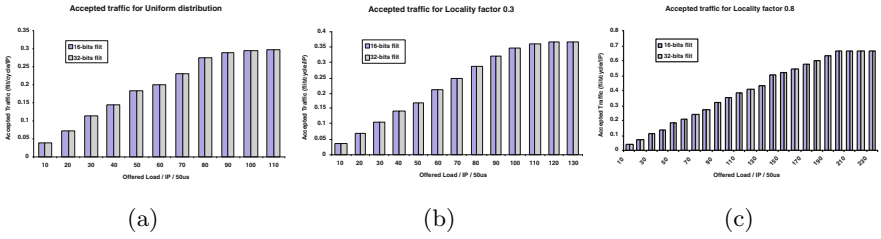


Fig. 2. Throughput variation profile for different locality factor

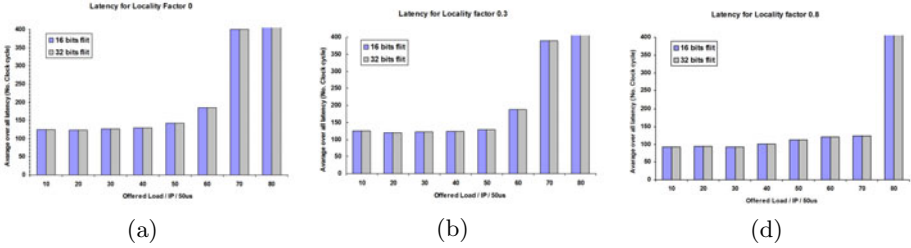


Fig. 3. Latency variation profile for different locality factor

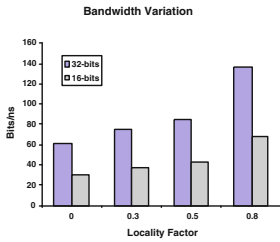


Fig. 4. Bandwidth variation

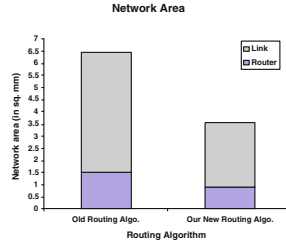


Fig. 5. Network area variation

10 Conclusion

In this paper we have proposed a simple and novel routing algorithm. For simplicity of routing algorithm we got reduction in area. This is expected to yield reduction in power. Due to the simplified addressing scheme, the minimum flit size could be reduced to 16-bits. We compared the performance of our work with the earlier work [2, 3, 4]. At minimum flit size, though we are losing bandwidth, it can be improved easily. Our design does not put any restriction. This flexibility will help us in SoC design, as we will be able to use reduced complexity routers with lower flit size for tasks demanding lesser bandwidth. For tasks demanding higher bandwidth, higher flit size can be utilized. The works reported in [2, 3, 4] lack this flexibility. We are currently working on obtaining the power profile for our NoC design.

References

- [1] Flich, J., Rodrigo, S., Duato, J.: An Efficient Implementation of Distributed Routing Algorithm for NoCs. In: Proceedings of Second ACM/IEEE International Symposium on Networks-on-Chip, pp. 87–96 (2008)
- [2] Kundu, S., Chattopadhyay, S.: Network-on-chip architecture design based on Mesh-of-tree deterministic routing topology. *Int'l Journal of High Performance System Architecture* 1(3), 163–182 (2008)
- [3] Kundu, S., Chattopadhyay, S.: Mesh-of-tree deterministic routing for network-on-chip architecture. In: In Proceedings of ACM Great Lakes Symposium on VLSI 2008, pp. 343–346 (2008)
- [4] Kundu, S., Manna, K., Gupta, S., Kumar, K., Parikh, R., Chattopadhyay, S.: A Comparative Performance Evaluation of Network-on-Chip Architectures under Self-Similar Traffic. In: Proceedings of International conference on Advances in Recent Technologies in communication and Computing, pp. 414–418 (2009)
- [5] Glass, C.J., Ni, L.M.: Turn Model for Adaptive Routing. In: Proceedings of the 19th annual international symposium on Computer architecture (ISCA 1992), pp. 278–287 (1992)
- [6] Balkan, A.O., et al.: A Mesh-of-Trees Interconnection Network for Single-Chip Parallel Processing. In: IEEE 17th International Conference on Application-specific Systems, Architectures and Processors, ASAP 2006 (2006)
- [7] Pande, P.P., Grecu, C., Jones, M., Ivanov, A., Saleh, R.: Performance Evaluation and Design Trade-Offs for Network-on-Chip Interconnect Architectures. In *IEEE Transaction on computers* 54(8), 1025–1040 (2005)

2-D DOA Estimation of Coherent Wideband Signals Using L-Shaped Sensor Array

P.M. Swetha¹ and P. Palanisamy²

¹ M.Tech student

² Assistant Professor, Electronics & Communication Engineering Department,
National Institute of Technology, Tiruchirapalli, TamilNadu, India-620015
swethapottekatt@gmail.com
palan@nitt.edu

Abstract. In this paper, a two-dimensional (2-D) direction-of-arrival (DOA) angle estimation method for coherent wideband signals using L-shaped array is introduced. This method estimates the elevation and azimuth angles using ESPRIT technique. The incident wideband signals are assumed to be coherent and hence some decorrelation techniques are employed to decorrelate them. Then the cross-correlation matrix is found. The proposed method reconstructs the signal subspace using cross-correlation matrix and then ESPRIT algorithm is employed to estimate the azimuth and elevation angles. This approach enables the 2-D DOA estimation without the use of peak search and is robust to noise. Simulation results are presented to verify the effectiveness of the proposed algorithm.

Keywords: Direction of arrival, Angle estimation, Wideband, Cross-correlation, Coherent signals, L-shaped sensor array, ESPRIT.

1 Introduction

In recent years, several techniques have been proposed to solve the problem of estimating the direction-of-arrivals (DOAs) of multiple wideband sources. Direct exploitation of raw wideband signals for DOA estimation using the traditional narrowband technique leads to failure. The reason is that the energy of narrowband signals are concentrated in a frequency band which is relatively small compared with the center frequency, thus the sensor output can be easily vectorized by using one frequency component. It exploits the fact that time delays directly translate to phase differences so long as the phase remains approximately constant over the bandwidth. In the case of wideband sources, the phase difference between sensor outputs is not only dependent on the DOA, but also dependent on the temporal frequency. For wideband signals, whose bandwidth is not small compared to the centre frequency, this direct proportionality between time delays and phase differences does not hold. The common method of processing wideband sources is to decompose the wideband sources into a set of narrowband signals of different frequency by DFT. Narrowband methods can then be applied to each narrowband component of the decomposed signal. Based on this method, many algorithms were introduced. The incoherent signal subspace method (ISSM) is one of the simplest wideband DOA estimation

methods. ISSM process each frequency bin independently and simply averages the results over all the frequency bins. Since each decomposed signal is approximated as a narrowband signal, any narrowband DOA estimation method is applicable. ISSM is simple and effective when the SNR is high, but it suffers when the SNR at each frequency varies because of the bad DOA estimate at some frequencies. In order to overcome these disadvantages, a number of improved methods have been proposed. Coherent signal subspace method(CSSM) [1-2], 2D DFT based method[3], weighted average of signal subspaces (WAVES)[4] and test of orthogonality of projected subspaces (TOPS) [5] are some of them. CSS method uses wideband focusing concept[6,7]. Some of the other important contributions to DOA estimation of broadband sources include the works of Agrawal and Prasad [8], P. Palanisamy, N. Kalyanasundaram and A. Raghunandan [9], K. Anoop Kumar, R. Rajagopal and P. Ramakrishna Rao [10].Wideband signals have inherent advantages over narrowband signals in applications such as radar, sonar, seismology and spread spectrum communications. For example, wideband signals are used to track moving objects from acoustic measurements or to find buried objects with the help of seismic sensors. Ultrawideband (UWB) radar can provide high resolution images, and UWB wireless communication can counteract the effects of fading due to multipath channels. Use of wideband signals results in high data rates in communication.

This paper aims to develop effective 2-D DOA estimation methods for coherent wideband signals using L-shaped pressure sensor array. First we decompose the wideband sources into a set of narrowband signals of different frequency by DFT. Focusing matrices are constructed to transform each of the narrowband covariance matrices into the one corresponding to the reference frequency bin which is referred to as focusing frequency. Then we apply decorrelation techniques to decorrelate the coherent incident signals and ESPRIT algorithm is employed to estimate the elevation & azimuth angles. No peak search is required in this case for DOA estimation. This work can be considered as an extension of effective 2-D DOA estimation method for narrowband coherent signals developed by Jian-Feng Gua, Ping Wei, Heng-Ming Tai[11].

2 Data Model

Consider two uniform linear orthogonal arrays that form the L-shape sensor array configuration in the x-z plane as shown in Fig. 1. Each linear array consists of M-1 sensor elements with spacing d and the element placed at the origin is common for referencing. Suppose that there are P wideband signals with wavelength λ impinging on the pressure sensor array from distinct directions. The i^{th} signal has an elevation angle θ_i and an azimuth angle ϕ_i , $i=1, 2, 3, \dots, P$. The received signals are assumed to be zero-mean, wide-sense stationary random processes band-limited to W over the finite observation interval. The observation interval is subdivided into D disjoint intervals. The received signal $\tilde{z}_m(t)$ at the m^{th} sensor, in each subinterval, along the z-axis can be written as

$$\tilde{z}_m(t) = \sum_{i=1}^P \tilde{s}_i(t - \tau_m(\theta_i)) + \tilde{n}_{zm}(t), \quad m=0, 1, 2, \dots, (M-1) \quad (1)$$

where $\tilde{s}_i(t)$ is the i^{th} source signal, $\tilde{n}_{zm}(t)$ is the noise observed at the m^{th} sensor, and $\tau_m(\theta_i) = \frac{(m-1)d \cos(\theta_i)}{c}$ is the relative delay where c is the constant propagation speed of the source signals.

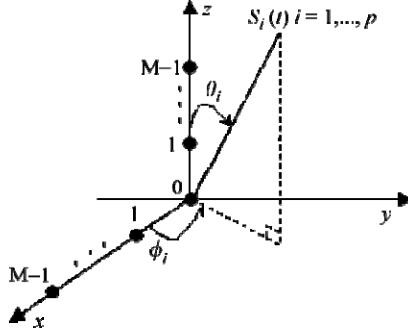


Fig. 1. Array element geometry configuration for the joint elevation and azimuth DOA estimation, ●: array element location

The sampled signal at the m^{th} sensor in each sub interval is decomposed into N narrowband components $\mathbf{z}_m(\omega_n)$, $n = 0, 1, 2, \dots, N-1$, using DFT. The resulting narrowband components corresponding to any one of the subintervals along the z -axis may be expressed as

$$\mathbf{z}_m(\omega_n) = \sum_{i=1}^P \mathbf{s}_i(\omega_n) \exp(-j\omega_n \tau_m(\theta_i)) + \mathbf{n}_{zm}(\omega_n) \quad (2)$$

where $\mathbf{s}_i(\omega_n)$ is the n^{th} frequency component of the source signal $\tilde{s}_i(t)$ and $\mathbf{n}_{zm}(\omega_n)$ is n^{th} frequency component of the noise process $\tilde{n}_{zm}(t)$.

Now, we can represent the received signal along the z -axis as

$$\mathbf{Z}(\omega_n) = \mathbf{A}_z(\omega_n) \mathbf{S}(\omega_n) + \mathbf{N}_z(\omega_n) \quad (3)$$

which consists of M rows.

$$\mathbf{A}_z(\omega_n) = [\mathbf{a}_z(\theta_1, \omega_n) \quad \mathbf{a}_z(\theta_2, \omega_n) \quad \dots \quad \mathbf{a}_z(\theta_p, \omega_n)]$$

is the $M \times P$ steering matrix at the n^{th} frequency f_n along z axis,

$$\mathbf{a}_z(\theta_i, \omega_n) = [1 \quad e^{-j2\pi \frac{d}{c} f_n \cos(\theta_i)} \quad e^{-j4\pi \frac{d}{c} f_n \cos(\theta_i)} \quad \dots \quad e^{-j2\pi(M-1) \frac{d}{c} f_n \cos(\theta_i)}]^T$$

$$\mathbf{S}(\omega_n) = [\mathbf{s}_1(\omega_n) \quad \mathbf{s}_2(\omega_n) \quad \dots \quad \mathbf{s}_p(\omega_n)]^T \text{ and}$$

$$\mathbf{N}_z(\omega_n) = [\mathbf{n}_{z,0}(\omega_n) \quad \mathbf{n}_{z,1}(\omega_n) \quad \dots \quad \mathbf{n}_{z,M-1}(\omega_n)]^T$$

Similarly, the resulting narrowband components corresponding to any one of the subintervals along the x -axis may be expressed as

$$\mathbf{x}_m(\omega_n) = \sum_{i=1}^P \mathbf{s}_i(\omega_n) \exp(-j\omega_n \tau_m(\theta_i, \phi_i)) + \mathbf{n}_{xm}(\omega_n) \quad (4)$$

where $\tau_m(\theta_i, \phi_i) = \frac{(m-1)d \sin(\theta_i) \cos(\phi_i)}{c}$ is the relative delay.

We can represent the received signal along the x-axis as

$$\mathbf{X}(\omega_n) = \mathbf{A}_x(\omega_n) \mathbf{S}(\omega_n) + \mathbf{N}_x(\omega_n) \tag{5}$$

which has M rows.

$$\mathbf{A}_x(\omega_n) = [\mathbf{a}_x(\theta_1, \phi_1, \omega_n) \quad \mathbf{a}_x(\theta_2, \phi_2, \omega_n) \quad \dots \quad \mathbf{a}_x(\theta_P, \phi_P, \omega_n)]$$

$$\mathbf{a}_x(\theta_i, \phi_i, \omega_n)$$

$$= [1 \quad e^{-j2\pi \frac{d}{c} f_n \sin(\theta_i) \cos(\phi_i)} \quad e^{-j4\pi \frac{d}{c} f_n \sin(\theta_i) \cos(\phi_i)} \quad \dots \quad e^{-j2\pi(M-1) \frac{d}{c} f_n \sin(\theta_i) \cos(\phi_i)}]^T$$

$$\text{and } \mathbf{N}_x(\omega_n) = [\mathbf{n}_{x,0}(\omega_n) \quad \mathbf{n}_{x,1}(\omega_n) \quad \dots \quad \mathbf{n}_{x,M-1}(\omega_n)]^T$$

The cross-covariance matrix $\tilde{\mathbf{R}}(\omega_n)$ at the n^{th} frequency f_n is given by

$$\begin{aligned} \tilde{\mathbf{R}}(\omega_n) &= E[\mathbf{Z}(\omega_n) \mathbf{X}^H(\omega_n)] \\ &= \mathbf{A}_z(\omega_n) \mathbf{R}_s(\omega_n) \mathbf{A}_x^H(\omega_n) + \mathbf{R}_n(\omega_n) \end{aligned} \tag{6}$$

which is a $M \times (M-1)$ matrix.

$\mathbf{R}_s(\omega_n)$ and $\mathbf{R}_n(\omega_n)$ are the covariance matrices of source signals and noise at the n^{th} frequency bin, respectively.

$\tilde{\mathbf{X}}(\omega_n)$ and $\tilde{\mathbf{A}}_x(\omega_n)$ are formed by taking the 2^{nd} to M^{th} rows of $\mathbf{X}(\omega_n)$ and $\mathbf{A}_x(\omega_n)$ respectively.

Superscript T denotes the transpose. We partition the incident signals into q groups. Signals in the same group are the delayed and scaled replica of each other, but orthogonal to signals in the other groups. It is assumed that both the number of signals P and the group number q are known or pre-estimated using some detection techniques, with $M-P > l_{\max}$, where $l_{\max} = \max\{l_1, l_2, \dots, l_q\}$ and l_k is the number of coherent signals in the k^{th} group. All signals $\mathfrak{S}_i(t)$ are uncorrelated with the additive noise $\{\tilde{\mathbf{n}}_{xm}(t), \tilde{\mathbf{n}}_{zm}(t)\}$ that are temporally white, zero-mean complex Gaussian random processes with second-order moments given by

$$\begin{aligned} \mathbf{Q} &= E\{\tilde{\mathbf{n}}_x(t) \tilde{\mathbf{n}}_x^H(t)\} = E\{\tilde{\mathbf{n}}_z(t) \tilde{\mathbf{n}}_z^H(t)\} \\ &= \text{diag}(\sigma_1^2, \sigma_2^2, \dots, \sigma_M^2) \end{aligned} \tag{7}$$

where $E\{\cdot\}$ denotes the statistical expectation and $(\cdot)^H$ the complex conjugate transpose.

2.1 Steps for Decomposing Wideband Sources into Narrowband Sources

Step 1: Divide the sensor output data $\mathbf{z}(t)$ and $\mathbf{x}(t)$ observed along the z-axis and x-axis respectively into D identically sized non-overlapping blocks of length L.

Step 2: Compute the temporal L-point DFT on each block d, $d=1,2,\dots,D$ and select N points having highest power concentrations from each block that corresponds to the N frequency bins present in the wideband signal. Thus we can form N narrowband signals each having D snapshots.

Step 3: Now apply narrowband techniques on the resulting narrowband signals.

3 Wideband Focusing for Pressure Sensor Array

In this section, we briefly outline the focusing method. The first step following the frequency decomposition of the array data vector is to align or focus the signal space at all frequency bins into a common bin at a reference frequency by focusing matrices $T(f_n)$ that satisfy

$$T(f_n)A(f_n) = A(f_o), n=1, 2, \dots, N \tag{8}$$

where f_o is some reference frequency called focusing frequency.

The array output along the z-axis after focusing transform is

$$Z(\omega_n) = T_z(\omega_n) \mathbf{Z}(\omega_n) \tag{9}$$

where $T_z(\omega_n)$ is the focusing matrix along the z-axis for the frequency ω_n .

Similarly, the array output along the x-axis after focusing transform is

$$Y_x(\omega_n) = T_x(\omega_n) \mathbf{X}(\omega_n) \tag{10}$$

where $T_x(\omega_n)$ is the focusing matrix along the x-axis for the frequency ω_n which is of order $(M-1) \times (M-1)$.

The cross-correlation matrix corresponding to the n^{th} frequency bin after focusing is

$$\begin{aligned} R_y(\omega_n) &= E[Y_z(\omega_n) Y_x^H(\omega_n)] \\ &= T_z(\omega_n) E[Z(\omega_n) \mathbf{X}^H(\omega_n)] T_x^H(\omega_n) \end{aligned} \tag{11}$$

$R_y(\omega_n)$ is a $M \times (M-1)$ matrix.

The coherently averaged cross-correlation matrix is given by

$$R_y = \sum_{n=1}^N R_y(\omega_n) \tag{12}$$

3 DOA Estimation Using ESPRIT Technique

If all of the P incoming signals are incoherent, then the rank of the cross-correlation matrix R_y equals the number of incident wideband signals P. Under this circumstance, we can extract any P columns from R_y to form a signal subspace. However, when the signals are coherent, we cannot create the signal subspace directly. Thus, the matrix decomposition technique is employed to decorrelate these coherent incident signals.

4.1 Elevation Angle estimation

4.1.1 Forward Cross-Correlation Estimation Method

To decorrelate the incident signals, first we partition the cross-correlation matrix R_y into l_{max} submatrices with the size of $(M - l_{max} + 1) \times (M - 1)$ each. The k^{th} submatrix is denoted as $R_y^{\{k\}}$, which is composed of k^{th} row to the $(M - l_{max} + k)^{\text{th}}$ row of R_y . Combining these submatrices together, we form R which is a matrix of order $(M - l_{max} + 1) \times (M - 1) l_{max}$.

Therefore,

$$\mathbf{R} = [\mathbf{R}_y^{\{1\}} \mathbf{R}_y^{\{2\}} \dots \dots \dots \mathbf{R}_y^{\{l_{max}\}}] \tag{13}$$

Now we can see that \mathbf{R} is having a rank \mathbf{P} , and it is possible to extract \mathbf{P} singular vectors associated with the nonzero singular values of \mathbf{R} to form the signal subspace \mathbf{U}_z .

4.1.2 Forward-Backward Cross-Correlation Estimation Method

Here, the forward cross-correlation estimation method is extended to the FB (Forward Backward) estimation method. The backward version of the matrix \mathbf{R} is defined as

$$\mathbf{R}_B = \mathbf{J}\mathbf{R}^* = \begin{bmatrix} 0 & \dots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \dots & 0 \end{bmatrix} \mathbf{R}^* \tag{14}$$

where \mathbf{J} is a $(M-l_{max}+1) \times (M-l_{max}+1)$ matrix. Using \mathbf{R}_B and \mathbf{R} , the FB cross-covariance matrix is

$$\mathbf{R}_{FB} = [\mathbf{R} \ \mathbf{R}_B] \tag{15}$$

The rank of \mathbf{R}_{FB} is \mathbf{P} . Applying the same procedure as in the forward method, the **FB** signal subspace $\mathbf{U}_{z,FB}$ is constructed using the \mathbf{P} singular vectors associated with the nonzero singular values of \mathbf{R}_{FB} and ESPRIT algorithm is then employed to estimate the elevation angles.

Form $\mathbf{U}_{s1} = \mathbf{J}_{s1} \mathbf{U}_s$ and $\mathbf{U}_{s2} = \mathbf{J}_{s2} \mathbf{U}_s$ where $\mathbf{J}_{s1} = [\mathbf{J}_s \ \mathbf{z}]$, $\mathbf{J}_{s2} = [\mathbf{z} \ \mathbf{J}_s]$, \mathbf{U}_s is the signal subspace.

$\mathbf{U}_s = \mathbf{U}_z$ for forward cross-correlation estimation

$\mathbf{U}_s = \mathbf{U}_{z,FB}$ for FB cross-correlation estimation

\mathbf{J}_s is an identity matrix of order $(M-l_{max})$

\mathbf{z} is a zero matrix of order $(M-l_{max}) \times 1$

Nonzero eigen values of $((\mathbf{U}_{s1})^H \mathbf{U}_{s1})^{-1} ((\mathbf{U}_{s1})^H \mathbf{U}_{s2})$ are found using which elevation angles θ_i can be calculated.

$$\hat{\theta}_i = \cos^{-1} \left(-\frac{\text{arg}(\gamma_{zi})}{2\pi d/\lambda} \right) \tag{16}$$

where γ_{zi} denotes the eigen values, $i=1,2,\dots,\dots,\mathbf{P}$.

4.2 Azimuth Angle Estimation

Using the estimated elevation angles, the array response vectors $\mathbf{A}_z(\omega_n)$ in the z-axis subarray can be obtained, which can be used to estimate the incident signals.

$$\begin{aligned} \mathbf{S}_z(\omega_n) &= (\mathbf{A}_z^H(\omega_n) \mathbf{A}_z(\omega_n))^{-1} \mathbf{A}_z^H(\omega_n) \mathbf{Z}(\omega_n) \\ &= (\mathbf{A}_z^H(\omega_n) \mathbf{A}_z(\omega_n))^{-1} \mathbf{A}_z^H(\omega_n) (\mathbf{A}_z(\omega_n) \mathbf{S}(\omega_n) + \mathbf{N}_z(\omega_n)) \\ &= \mathbf{S}(\omega_n) + (\mathbf{A}_z^H(\omega_n) \mathbf{A}_z(\omega_n))^{-1} \mathbf{A}_z^H(\omega_n) \mathbf{N}_z(\omega_n) \\ \mathbf{S}_z(\omega_n) &= \mathbf{S}(\omega_n) + \mathbf{N}_{Az}(\omega_n) \end{aligned} \tag{17}$$

where $\mathbf{N}_{Az}(\omega_n)$ is the additive noise.

Cross-correlation matrix between the received data of x-axis subarray and the estimated signals $\mathbf{S}_z(\omega_n)$ is calculated as follows.

$$\mathbf{R}_{x,s}(\omega_n) = \mathbf{E}\{\mathbf{X}(\omega_n) \mathbf{S}_z^H(\omega_n)\} \tag{18}$$

Using the same procedure as we followed earlier, focus the signal space at all frequency bins into a common bin at a reference frequency by focusing matrices. Let $\mathbf{R}_x(\omega_n)$ be the cross-correlation matrix corresponding to the n^{th} frequency bin after focusing transform. The coherently averaged cross-correlation matrix is given by

$$\mathbf{R}_x = \sum_{n=1}^N \mathbf{R}_x(\omega_n) \tag{19}$$

By applying the similar procedure as described before, we define the matrices $\tilde{\mathbf{R}}_x$ and $\tilde{\mathbf{R}}_{x_{FB}}$ as

$$\tilde{\mathbf{R}}_x = [\mathbf{R}_x^{\{1\}} \ \mathbf{R}_x^{\{2\}} \ \dots \ \dots \ \dots \ \dots \ \mathbf{R}_x^{\{lmax\}}] \tag{20}$$

Using $\tilde{\mathbf{R}}_{x_{FB}}$ and $\tilde{\mathbf{R}}_x$, the FB cross-correlation matrix is

$$\tilde{\mathbf{R}}_{x_{FB}} = [\tilde{\mathbf{R}}_x \ \tilde{\mathbf{R}}_{x_{FB}}] \tag{21}$$

Where
$$\tilde{\mathbf{R}}_{x_{FB}} = \mathbf{J} \tilde{\mathbf{R}}_x^* = \begin{bmatrix} \mathbf{0} & \dots & \mathbf{1} \\ \vdots & \ddots & \vdots \\ \mathbf{1} & \dots & \mathbf{0} \end{bmatrix} \tilde{\mathbf{R}}_x^* \tag{22}$$

Similar procedure as explained before for the estimation of elevation angles from \mathbf{R} or \mathbf{R}_{FB} is applied for the estimation of azimuth angles from $\tilde{\mathbf{R}}_x$ or $\tilde{\mathbf{R}}_{x_{FB}}$. P singular vectors associated with nonzero singular values of $\tilde{\mathbf{R}}_x$ or $\tilde{\mathbf{R}}_{x_{FB}}$ are extracted to form the signal subspace and then ESPRIT algorithm is employed to estimate the azimuth angles.

$$\hat{\phi}_i = \cos^{-1} \left(-\frac{\arg(\gamma_{xi})}{2\pi d (\sin \theta_i) / \lambda} \right) \tag{23}$$

where γ_{xi} denotes the eigen values, $i=1,2,\dots,P$.

5 Simulations and Results

In this section, simulation results are presented to demonstrate the effectiveness of the proposed method in the 2-D DOA estimation of coherent wideband signals. We evaluate the performance of pressure ESPRIT for coherent wideband signals. The following simulation condition assumptions are made for the tests: The sensor displacement d between adjacent elements in each uniform linear array is equal to half the wavelength of the signal wave. We consider an L shaped pressure sensor array which consists of $2M-1$ sensors in total, ie, M sensors along the z and x -axes including the common reference sensor element. The signals of interest consisted of 5 narrowband components lying in the frequency band 80-120Hz. Then focusing technique is applied. The focusing frequency is chosen as $f_0 = 80\text{Hz}$. The total 3000 successive data samples at each sensor are collected for each simulation trial, which are divided into 15 segments, each having 200 samples (that is, $D=15, L=200$), to estimate the covariance matrices. The joint elevation and azimuth DOA estimation criterion is the root mean square error (RMSE) defined as

$$\text{RMSE} = \sqrt{E[(\hat{\theta} - \theta)^2 + (\hat{\phi} - \phi)^2]}$$

where $\hat{\theta}, \hat{\phi}$ are the estimated elevation & azimuth angles and θ, ϕ are the actual elevation & azimuth angles respectively.

In the first test, we consider two coherent signals with identical powers (that is, $P=2, q=1$ and $l_{\max}=2$) and different DOAs $(\theta, \phi) = (50^\circ, 40^\circ)$ and $(70^\circ, 60^\circ)$. Here we evaluate the RMSE performance of the ESPRIT algorithm at various SNRs for wideband signals. Fig.2 shows the RMSE plots for the joint elevation and azimuth DOA estimation for two signals of DOAs $(50^\circ, 40^\circ)$ and $(70^\circ, 60^\circ)$ versus the SNR. Other parameters include the SNR varying from -10dB to 30dB, number of snapshots =200, total number of sensor elements is 7, ie., $2M-1=7, M=4$ and 100 independent runs. Results show that RMSE decreases as the SNR increases. It can be concluded from Fig.2 that the forward backward (FB) method gives better performance in DOA estimation compared to the forward method. This also demonstrates the fact that, the use of cross-correlation matrix and the FB averaging effectively improves the degree of decorrelation and the estimation performance, respectively.

In the second test, we evaluate the performance in terms of Cluster Diagram. Cluster diagram is the plot of elevation angle versus azimuth angle. Simulation conditions are same as that of the first test except that SNR is fixed at 10dB. Cluster diagrams shown in the Fig.3 ensures the fact that Forward Backward(FB) method gives better performance in the estimation of DOA compared to Forward ESPRIT method.

In the third test, we evaluate the performance in terms of Detection Probability. Simulation conditions are same as that in the first test. It can be seen from fig.4 that the FB ESPRIT method exhibits better detection probability compared to Forward ESPRIT method.

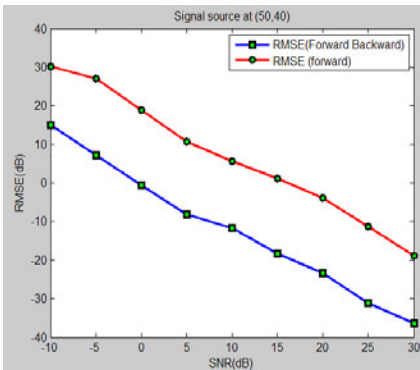


Fig. 2(a)

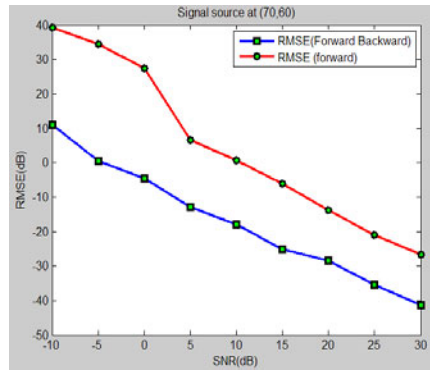


Fig. 2(b)

Fig. 2. RMSE plots for the joint elevation and azimuth DOA estimation for two signals of DOAs a) $(50^\circ, 40^\circ)$ and b) $(70^\circ, 60^\circ)$ versus SNR

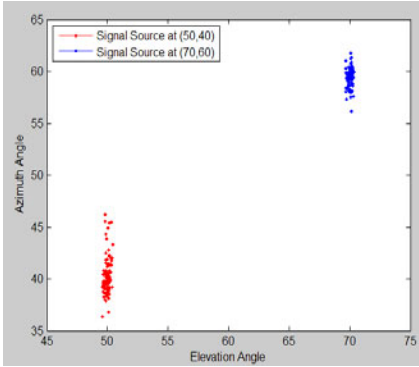


Fig. 3(a)

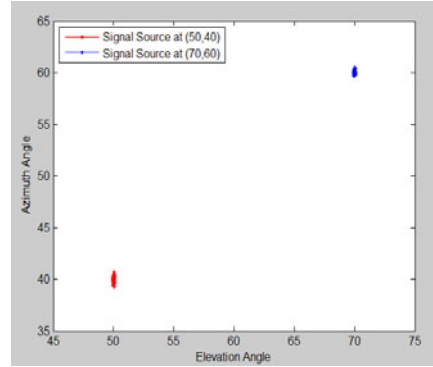


Fig. 3(b)

Fig. 3. Cluster Diagram for a) forward ESPRIT method b) forward backward ESPRIT method

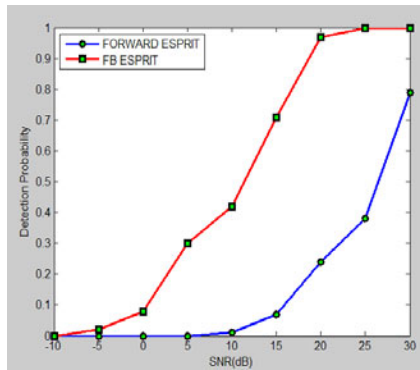


Fig. 4. Detection Probability versus SNR for two coherent signal sources at (50°, 40°) and (70°,60°)

6 Conclusion

In this paper we studied the behavior of the ESPRIT method using pressure sensors in the 2-D Direction-of-Arrival estimation of coherent wideband signals. The proposed forward and FB methods construct a signal subspace by decomposing the cross-covariance matrix and employing the shift invariance property. The proposed techniques offers closed form automatically paired DOA estimation, which do not require any iterative peak search for estimating DOA parameter. Simulation results have demonstrated that the proposed FB method exhibits superior performance for DOA estimation of coherent signals, especially at low SNR and small number of snapshots.

References

- [1] Wang, H., Kaveh, M.: Coherent Signal-Subspace Processing for the Detection and Estimation of Angles of Arrival of Multiple Wide-Band Sources. *IEEE Transactions on Acoustics, Speech, and Signal Processing* ASSP-33(4) (August 1985)
- [2] Chen, H., Zhao, J.: Coherent signal-subspace processing of acoustic vector sensor array for DOA estimation of wideband sources. *Signal Processing* 85, 837–847 (2005)
- [3] Allam, M., Moghaddamjoo, A.: Two-Dimensional DFT Projection for Wideband Direction-of-Arrival Estimation. *IEEE Transactions on Signal Processing* 43, 1728–1732 (1995)
- [4] di Claudio, E.D., Parisi, R.: WAVES: weighted average of signal subspaces for robust wideband direction finding. *IEEE Transactions on Signal Processing* 49, 2179–2191 (2001)
- [5] Yoon, Y.-S., Kaplan, L.M., McClellan, J.H.: TOPS: New DOA Estimator for Wideband Signals. *IEEE transactions on Signal Processing* 54, 1977–1989 (2006)
- [6] Hung, H., Kaveh, M.: Focusing Matrices for Coherent Signal-Subspace Processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 36(8) (August 1988)
- [7] Sellone, F.: Robust auto-focusing wideband DOA estimation. *Signal Processing* 86, 17–37 (2006)
- [8] Agrawal, M., Prasad, S.: Estimation of directions of arrival of wideband and wideband spread sources. *Signal Processing* 87, 614–622 (2007)
- [9] Palanisamy, P., Kalyanasundaram, N., Raghunandan, A.: A new DOA estimation algorithm for wideband signals in the presence of unknown spatially correlated noise. *Signal Processing* 89, 1921–1931 (2009)
- [10] Anoop Kumar, K., Rajagopal, R., Ramakrishna Rao, P.: Wide-band DOA estimation in the presence of correlated noise. *Signal Processing* 52, 23–34 (1996)
- [11] Gu, J.-F., Wei, P., Tai, H.M.: 2-D direction-of-arrival estimation of coherent signals using cross-correlation matrix. *Signal Processing* 88, 75–85 (2008)

An Approach towards Lightweight, Reference Based, Tree Structured Time Synchronization in WSN

Surendra Rahamatkar¹ and Ajay Agarwal²

¹ Dept. of Computer Science, Nagpur Institute of Technology, Nagpur, India

² Dept. of MCA, Krishna Institute of Engg. & Technology, Ghaziabad, India

Abstract. Time synchronization for wireless sensor networks (WSNs) has been studied in recent years as a fundamental and significant research issue. Many applications based on these WSNs assume local clocks at each sensor node that need to be synchronized to a common notion of time. Time synchronization in a WSN is critical for accurate time stamping of events and fine-tuned coordination among the sensor nodes to reduce power consumption. This paper proposes a lightweight tree structured time synchronization approach for WSNs based on reference nodes. This offers a push mechanism for (i) accurate and (ii) low overhead for global time synchronization. Analysis and comparative study of proposed approach shows that it is lightweight as the number of required broadcasting messages is constant in one broadcasting domain.

Keywords: Reference Node, Tree Structure Topology, Time Synchronization, Wireless Sensor Network.

1 Introduction

Wireless sensor networks (WSNs) can be applied to a wide range of applications in domains as diverse as medical, industrial, military, environmental, scientific, and home networks [3],[4].

Time synchronization of WSNs is crucial to maintain data consistency, coordination, and perform other fundamental operations. Further, synchronization is considered a critical problem for wireless ad hoc networks due to its de-centralized nature and the timing uncertainties introduced by the imperfections in hardware oscillators and message de-lays in both physical and medium access control (MAC) layers. All these uncertainties cause the local clocks of different nodes to drift away from each other over the course of a time interval.

The clock synchronization problem has been studied thoroughly in the areas of Internet and local area networks (LANs) for the last several decades. Many existing synchronization algorithms rely on the clock information from Global Positioning System (GPS). However, GPS-based clock acquisition schemes exhibit some weaknesses: GPS is not ubiquitously available and requires a relatively high-power receiver, which is not possible in tiny and cheap sensor nodes. This is the motivation for developing software-based approaches to achieve in-network time synchronization.

Among many protocols that have been devised for maintaining synchronization, Network Time Protocol (NTP) [5] is outstanding owing to its ubiquitous deployment, scalability, robustness related to failures, and self-configuration in large multi-hop networks. Moreover, the combination of NTP and GPS has shown that it is able to achieve high accuracy on the order of a few microseconds [6]. However, NTP is not suitable for a wireless sensor environment, since WSNs pose numerous challenges of their own; to name a few, limited energy and bandwidth, limited hardware, latency, and unstable network conditions caused by mobility of sensors, dynamic topology, and multi-hopping. The most of the time synchronization protocols differ broadly in terms of their computational requirements, energy consumption, precision of synchronization results, and communication requirements [1].

In the paper, we propose a more effective, lightweight multi-hop tree structured referencing time synchronization (TSRT) approach with the goal of achieving a long-term network-wide synchronization with minimal Message Exchanges and exhibits a number of attractive features such as highly scalable and lightweight.

The whole paper is organized in five Sections. In Section 2, existing synchronization schemes are reviewed. Proposed reference based tree structured time synchronization scheme is explained in Section 3. Section 4 contains the analysis of proposed scheme with existing work. Finally, Section 5 contains the conclusion of the paper.

2 Existing Approaches to Time Synchronization

Time synchronization algorithms providing a mechanism to synchronize the local clocks of the nodes in the network have been extensively studied in the past. Two of the most prominent examples of existing time synchronization protocols developed for the WSN domain are the Reference Broadcast Synchronization (RBS) algorithm [7] and the Timing-sync Protocol for Sensor Networks (TPSN) [8].

In RBS, a reference message is broadcasted. The receivers record their local time when receiving the reference broadcast and exchange the recorded times with each other. The main advantage of RBS is that it eliminates transmitter-side non-determinism. The disadvantage of the approach is that additional message exchange is necessary to communicate the local time-stamps between the nodes. In the case of multi hop synchronization, the RBS protocol would lose its accuracy. Santashil PalChaudhuri et al [13] extended the RBS protocol to handle multi hop clock synchronization in which all nodes need not be within single-hop range of a clock synchronization sender.

The TPSN algorithm first creates a spanning tree of the network and then performs pair wise synchronization along the edges. Each node gets synchronized by exchanging two synchronization messages with its reference node one level higher in the hierarchy. The TPSN achieves two times better performance than RBS by time-stamping the radio messages in the MAC layer of the radio stack and by relying on a two-way message exchange. The shortcoming of TPSN is that it does not estimate the clock drift of nodes, which limits its accuracy, and does not handle dynamic topology changes.

TinySeRSync [14] protocol works with the ad hoc deployments of sensor networks. This protocol proposed two asynchronous phases: Phase I– secure single-hop pair wise synchronization, and Phase II– secure resilient global synchronization to achieve global time synchronization in a sensor network.

Van Greunen et al [16] lightweight tree-based synchronization (LTS) protocol is a slight variation of the network-wide synchronization protocol of Ganeriwal et al. [15]. Similar to network-wide synchronization the main goal of the LTS protocol is to achieve reasonable accuracy while using modest computational resources. As with network-wide synchronization, the LTS protocol seeks to build a tree structure within the network. Adjacent tree nodes exchange synchronization information with each other. A disadvantage is that the accuracy of synchronization decreases linearly in the depth of the synchronization tree (i.e., the longest path from the node that initiates synchronization to a leaf node). Authors discuss various ideas for limiting the depth of tree; the performance of protocol is analyzed with simulations.

The clock synchronization protocols significantly differ from the conventional protocols in dealing the challenges specific to WSNs. It is quite likely that the choice of a protocol will be driven by the characteristics and requirements of each application.

3 TSRT Approach

In this Section we propose a Tree Structured Referencing Time Synchronization (TSRT) scheme, which is based on the protocol, proposed by [2], that the aim is to minimize the complexity of the synchronization. Thus the needed synchronization accuracy is assumed to be given as a constraint, and the target is to devise a synchronization algorithm with minimal complexity to achieve given precision. TSRT works on two phases. First phase used to construct an ad hoc tree structure and second phase used to synchronize the local clocks of sensor nodes followed by network evaluation phase.

The goal of the TSRT is to achieve a network wide synchronization of the local clocks of the participating nodes. We assume that each node has a local clock exhibiting the typical timing errors of crystals and can communicate over an unreliable but error corrected wireless link to its neighbors. The TSRT synchronizes the time of a sender to possibly multiple receivers utilizing a single radio message time-stamped at both the sender and the receiver sides. MAC layer time-stamping can eliminate many of the errors, as observed in [9]. However, accurate time-synchronization at discrete points in time is a partial solution only. Compensation for the clock drift of the nodes is inevitable to achieve high precision in-between synchronization points and to keep the communication overhead low. Linear regression is used in TSRT to compensate for clock drift as suggested in [7].

3.1 Main Ideas

The proposed synchronization approach is flexible and self-organized. A physical broadcast channel is required, which is automatically satisfied by the wireless medium. A connected network is also required in order to spread the synchronization ripple to nodes network wide. The proposed approach assumes the coexistence of reference nodes and normal sensor nodes in a WSN. A “*reference node*” periodically

transmits beacon messages to its neighbors. These beacon messages initiate the synchronization waves. Multiple reference nodes are allowed to operate in the system simultaneously. A sensor node in this approach will dynamically select the nearest reference node as its reference for clock synchronization.

The proposed protocol used for multi-hop synchronization of the network based on pair wise synchronization scheme suggested by [8]. An explanation of a standard two-way message exchange between a pair of nodes [11] employing for Synchronization is helpful to understand proposed synchronization design. The basic building block of the synchronization process is the two-way message exchange between a pair of nodes Here we assume that the clock drift between a pair of nodes is constant in the small time period during a single message exchange. The propagation delay is also assumed to be constant in both directions. Consider a two-way message exchange between nodes A and B as shown in Fig. 1. Node A initiates the synchronization by sending a synchronization message at time t_1 as per node's local clock. This Message includes A's identity, and the value of t_1 . B receives this message at t_2 which can be calculated as

$$t_2 = t_1 + \Delta + d \quad (1)$$

Where Δ is the relative clock drift between the nodes, and d is the propagation delay of the pulse. B responds at time t_3 with an acknowledgement, which includes the identity of B and the values t_1 , t_2 , and t_3 . Then, node A can calculate the clock drift and propagation delay as below, and synchronize itself with respect to node B.

$$\Delta = ((t_2 - t_1) - (t_4 - t_3))/2 \quad (2)$$

$$d = ((t_2 - t_1) + (t_4 - t_3))/2 \quad (3)$$

The synchronization phase is initiated by the root node's `syn_begin` message. On receiving this message, nodes of level 1 initiate a two-way message exchange with the root node. Before initiating the message exchange, each node waits for some random time, in order to minimize collisions on the wireless channel. Once they get back a reply from the root node, they adjust their clocks to the root node. Level 2 nodes, overhearing some level 1 node's communication with the root, initiate a two-way message exchange with a level 1 node, again after waiting for some random time to ensure that level 1 nodes have completed their synchronization. This procedure eventually gets all nodes synchronized with reference to the root node, the synchronization process described in detail in Subsection 3.3.

3.2 Ad Hoc Tree Construction Phase

Before synchronization, proposed Algorithm 1 is used by each sensor node to efficiently flood the network to form a logical hierarchical structure from a designated source point. Each sensor is initially set to accept `fd_pkt` (flood packets) for first time, but will ignore subsequent ones in order not to be continuously reassigned as the flood broadcast propagates. When a node receives or accepts the `fd_pkt` then first it

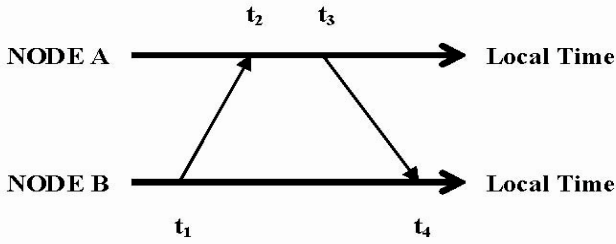


Fig. 1. Two way message exchange between a pair of nodes

set to its parent as source of broadcast after that level of current receiver node will be assigned one more than the level of parent node and then it broadcast the `fd_pkt` along with node identifier and level. If a node receives the `ack_pkt`, the variable `no_receiver` increments to maintain track of the node's receivers.

Algorithm 1. Ad hoc Tree Structure Construction

```

begin
  Accept (fd_pkts)
  Initialize : no_reciever = 0;
  Node_Level(Root)=0;
  If (current_reciever == root)
    Broadcast (fd_pkts)
    else if (current_reciever != root)
      begin
        Accept (fd_pkts);
        Parent(curent_reciever)= Source(broadcast_msg) ;
        Node_Level (curent_reciever)=
        Node_Level(Parent)+1;
        Broadcast (ack_pkt, node_id);
        Ignore (fd_pkts);
      end
    else if (current_node receives ack_pkt)
      no_receiver++;
end

```

3.3 Hierarchical Time Synchronization Phase

The first component of TSRT's bidirectional time synchronization service is the push-based [10] Hierarchy Time Synchronization (HTS) Scheme. The goal of HTS is to enable central authorities to synchronize the vast majority of a WSN in a lightweight manner. This approach particularly based on pair wise synchronization with allusion to single reference node is discussed in Subsection 3.1.

3.3.1 Single Reference Node

As shown in Fig. 2, HTS consists of three simple steps that are repeated at each level in the hierarchy. First, a Reference Node (RN) broadcasts a beacon on the control

channel (Fig. 2A). One child node specified by the reference node will jump to the specified clock channel, and will send a reply on the clock channel (Fig. 2B). The RN will then calculate the clock offset and broadcast it to all child nodes, synchronizing the first ripple of child nodes around the reference node (Fig. 2C). This process can be repeated at subsequent levels in the hierarchy further from the reference node (Fig. 2D). The HTS scheme is explained in more detail as follows:

Step 1: RN initiates the synchronization by broadcasting the `syn_begin` message with time t_1 using the control channel and then jumps to the clock channel. All concerned nodes record the received time of the message announcement. RN randomly specifies one of its children, e.g. SN2, in the announcement. The node SN2 jumps to the specified clock channel.

Step 2: At time t_3 , SN2 replies to the RN with its received times t_2 and t_3 .

Step 3.1: RN now contains all time stamps from t_1 to t_4 . It calculates clock drift Δ and propagation delay d , as per equation (2) and (3), and calculate $t_2 = t_1 + \Delta + d$, and then broadcasts it on the control channel.

Step 3.2: All involved neighbor nodes, (SN2, SN3, SN4 and SN5) compare the time t_2 with their received timestamp t_2' i.e. SN3 calculates the offset d' as:

$$d' = t_2 - t_2'$$

Finally, the time on SN3 is corrected as:

$$T = t + d + d'$$

Where t is the local clock reading.

Step 4: SN2, SN3, SN4 and SN5 initiate the `syn_begin` to their downstream nodes.

We assume that each sensor node knows about its neighbors when it initiates the synchronization process. In Step 1, the response node is specified in the announcement. It's the only node that jumps to the clock channel specified by the RN. The other nodes are not disturbed by the synchronization conversation between RN and SN2 and can conduct normal data communication while waiting for the second update from the RN. A timer is set in the RN when the `syn_begin` message is transmitted. In case the message is lost on its way to SN2, the RN goes back to the normal control channel after the timer expires and thus avoids indefinite waiting.

As the synchronization ripple spreads from the reference node to the rest of the network, a multi-level hierarchy is dynamically constructed. Levels are assigned to each node based on its distance to reference node, i.e. number of hops to the central reference point. Initiated from the reference nodes, the synchronization steps described above are repeated by the nodes on each level from the root to the leaves. To avoid being updated by redundant synchronization requests from peers or downstream nodes, HTS allows the nodes to parameterize their requests with two variables, i.e. "level" and "depth":

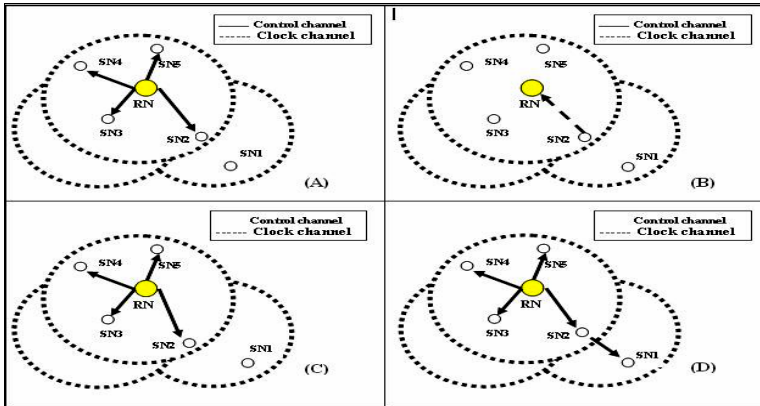


Fig. 2. (A) Reference node broadcasts (B) A neighbor replies (C) All neighbors are synchronized (D) Repeat at lower layers

A “level” indicating the number of hops to the synchronization point is contained in each `syn_begin` packet. At the very beginning of each synchronization ripple, the reference nodes set the level to 0 in the `syn_begin` packet. If a node M is updated by a `syn_begin` packet marked by level n , it will set its level to $n+1$ and then broadcast the `syn_begin` message to all its neighbors with level $n+1$. If M receives another `syn_begin` packet later during the same synchronization period, M will look into the “level” contained in this packet. If the new level is equal to or larger than n , then M will just ignore this updating request. Otherwise, it will respond to this `syn_begin` packet and update itself based on the sender. Following this procedure, a tree is constructed dynamically with the reference nodes sitting at the base. Each node is associated with a level according to its distance to the base. For example, in Fig. 2, the RN is at level 0, while all its neighbors SN2, SN3, SN4, and SN5 are at level 1 after being synchronized with the RN. After being updated by RN, the nodes SN2, SN3, SN4 and SN5 initiate the `syn_begin` packet to their child nodes. However, SN2 and SN3 are in each other’s broadcasting domain. SN3 will find SN2 to be at the same level, and therefore will simply ignore the synchronization request from SN2.

Besides “level”, HTS also allows network nodes to specify the radius of the synchronization ripple. The nodes could parameterize the request with a second field called “depth” in the `syn_begin` message. The initiating nodes set the “depth” field in the `syn_begin` packets. The value of “depth” is decremented by one in each level. The synchronization ripple stops spreading when the depth becomes zero. However, the downstream nodes could adjust the “depth” field according to their needs. With this flexible mechanism, the reference point will have the ability to control the range of the nodes that are updated.

4 Analysis of TSRT

The TSRT protocol exploits the broadcast nature of the wireless medium to establish a single common point of reference, i.e. the arrival time of the broadcast message is

the same on all neighbor peers. This common reference point can be used to achieve synchronization in Step 4 of the Subsection 3.3, i.e. t_2 at node SN2 occurred at the same instant as t_2' at node SN3. As the RN is synchronizing itself with SN2, the other neighboring nodes can overhear the RN's initial broadcast to SN2 as well as the RN's final update informing SN2 of its offset d_2 . If in addition the RN includes the time t_2 in the update sent to SN2 (redundant for SN2), then that allows all neighbors to synchronize.

The intuition is that, since t_2 and t_2' occurred at the same instant, then overhearing t_2 gives SN3 its offset from SN2's local clock and overhearing d_2 gives SN3 the offset from SN2's local clock to the RN reference clock. Thus, SN3 and all children of the RN can calculate their own offsets to the RN reference clock with only three messages (2 control broadcasts and 1 clock channel reply) TSRT is highly scalable and lightweight, since there is only one lightweight overhead exchange per hop between a parent node and all of its children. In contrast, synchronization in RBS happens between a pair of neighbors, which is called pair verification, rather than between a central node and all of its neighbors.

As a result, RBS is susceptible to high overhead as the number of peers increases [1]. The TSRT approach eliminates the potential broadcast storm that arises from pair wise verification, while at the same time preserving the advantage of reference broadcasting, namely the common reference point. Also, since the TSRT parent provides the reference broadcast that is heard by all children, then TSRT avoids the problem in RBS when two neighbors of an initiating peer are "hidden" from each other. The parameters used in the protocol dynamically assign the hierarchy level to each node during the spread of the synchronization ripple and no additional routing protocol is required. TSRT is lightweight since the number of required broadcasting messages is constant in one broadcasting domain. Only three broadcast messages are necessary for one broadcasting domain, no matter how dense the sensor nodes are. In TSRT, the sender error is eliminated by comparing the received time on each node. The major error sources come from:

Variance in the propagation delay: TSRT ignores the difference between the propagation time to different neighbors. As is illustrated in Fig. 1, node RN broadcasts to its neighbor's SN1 and SN2. The propagation time needed for the message to arrive at SN1 and SN2 are t_1 and t_2 , which are different in reality. As the propagation speed of electromagnetic signals through air is close to the speed of light, then for sensor network neighbors that are within tens of feet, the propagation time is in the nanosecond level and thus could be ignored compared with the other error sources. Also, TSRT makes the assumption that the propagation speeds are the same in different channels.

Receiver error: Latency due to processing in the receiver side is attributable to operating system jitter. However, sensor operating systems can be designed so that this jitter becomes relatively small and deterministic, e.g. the time can be read immediately at each interrupt generated by the radio interface when a packet is received.

TSRT's current policy for selecting the child node to respond to the sync begin message is a random selection. However, it is possible to incorporate historical knowledge from previous TSRT cycles in the selection of the next child responder. Moreover, previous TSRT responses may be combined to broadcast a composite value in Step 3. This may be useful to account for propagation delay differences

between neighbors within a local broadcast domain, which we can assumed to be small, but which may become more relevant when the distances between neighbors becomes very large in a highly distributed Wireless Sensor Network.

5 Conclusion

WSN have tremendous advantages for monitoring object movement and environmental properties but require some degree of synchronization to achieve the best results. The proposed TSRT synchronization approach is able to produce deterministic synchronization with only few pair wise message exchanges.

While the proposed approach is especially useful in WSN which are typically, extremely constrained on the available computational power, bandwidth and have some of the most exotic needs for high precision synchronization. The proposed synchronization approach was designed to switch between TPSN and RBS. These two algorithms allow all the sensors in a network to synchronize themselves within a few microseconds of each other, while at the same time using the least amount of resources possible. In this work two varieties of the algorithm are presented and their performance is verified theoretically with the existing results and compared with existing protocols. The comparison with existing approaches shows that the proposed synchronization approach is lightweight since the number of required broadcasting messages is constant in one broadcasting domain and synchronization is done using a few pair wise message exchanges.

References

1. Rahamatkar, S., Agarwal, A., Kumar, N.: Analysis and Comparative Study of Clock Synchronization Schemes in Wireless Sensor Networks. *Int. J. Comp. Sc. & Engg.* 2(3), 523–528 (2010)
2. Rahamatkar, S., Agarwal, A., Sharma, V.: Tree Structured Time Synchronization Protocol in Wireless Sensor Network. *J. Ubi. Comp. & Comm.* 4, 712–717 (2009)
3. Rhee, K., Lee, J., Wu, Y.C.: Clock Synchronization in Wireless Sensor Networks: An Overview. *Sensors* 9, 56–85 (2009)
4. Zhao, F., Guibas, L.: *Wireless Sensor Networks: An Information Processing Approach*, pp. 107–117. Morgan Kaufmann, San Francisco (2004)
5. Mills, D.L.: Internet Time Synchronization: The Network Time Protocol. *IEEE Trans. Comm.* 39(10), 1482–1493 (1991)
6. Bulusu, N., Jha, S.: *Wireless Sensor Networks: A Systems Perspective*. Artech House, Norwood (2005)
7. Elson, J.E., Girod, L., Estrin, D.: Fine-Grained Network Time Synchronization using Reference Broadcasts. In: *5th Symposium on Operating Systems Design and Implementation*, pp. 147–163 (2002)
8. Ganeriwal, S., Kumar, R., Srivastava, M.B.: Timing-Sync Protocol for Sensor Networks. In: *First ACM Conference on Embedded Networked Sensor System (SenSys)*, pp. 138–149 (2003)
9. Woo, A., Culler, D.: A Transmission Control Scheme for Media Access in Sensor Networks. In: *Mobicom*, pp. 221–235 (2001)

10. Dai, H., Han, R.: TSync: a lightweight bidirectional time synchronization service for wireless sensor networks. *SIGMOBILE Mob. Comput. Commun. Rev.* 8(1), 125–139 (2004)
11. Noh, K.L., Chaudhari, Q., Serpedin, E., Suter, B.: Analysis of clock offset and skew estimation in timing sync protocol for sensor networks. *IEEE Globecom*, San Francisco (2006)
12. Noh, K.-L., Serpedin, E.: Adaptive multi-hop timing synchronization for wireless sensor Networks. In: *9th Int. Symp. On Signal Processing & Its App.* (2007)
13. PalChaudhuri, S., Saha, A.K., Johnson, D.B.: Adaptive clock synchronization in sensor networks. In: *3rd Int. Symp. on Inf. Processing in Sensor Networks*, USA, pp. 340–348 (2004)
14. Sun, K., Ning, P., Wang, C.: TinySeRSync: Secure and Resilient time synchronization in wireless sensor networks. In: *13 ACM Conf. on Comp. Comm. Security*, USA, pp. 264–277 (2006)
15. Ganeriwal, S., Kumar, R., Adlakha, S., Srivastav, M.: Network-wide Time Synchronization in Sensor Networks. Technical Report, Networked and Embedded Systems Lab, Elec. Eng. Dept., UCLA (2003)
16. Greunen, J., Rabaey, J.: Lightweight Time Synchronization for Sensor Networks. In: *2nd ACM Int. Workshop on WSN & Applications*, California, pp. 11–19 (2003)

Towards a Hierarchical Based Context Representation and Selection by Pruning Technique in a Pervasive Environment

B. Vanathi and V. Rhymend Uthariaraj

Ramanujan Computing Centre, Anna University Chennai,
Chennai, Tamil Nadu, India
mbvanathi@yahoo.co.in, rhymend@annauniv.edu

Abstract. Pervasive computing objective is to merge computing and computing applications into surroundings instead of having computers as discrete objects. Applications must adjust their behavior to every changing surroundings. Adjustment involves proper capture, management and reasoning of context. Context is represented in a hierarchical form and stored in an object relational database. Context is selected using heuristic pruning method in this paper. Semantic of the context and context data is handled by Object Relational Database. Allowing only limited amount of context data to the reasoning space improves the performance of the reasoning process.

Keywords: Context Representation, Ontology, Relational Database, Object Relational Database, Pruning method.

1 Introduction

The advancement of computing applications goes with the evolution of distributed middleware. In e-commerce and cooperative business, the Web and its underlying protocols such as HTTP, SOAP, FTP etc are becoming the standard execution platform for distributed and component based applications. The increasing number of computers and users on the Internet has led not only to cooperative structures such as peer-to-peer computing that has great potential for scalability but also stimulated new developments in the area of computing clusters called grid computing. The integration of mobile clients into a distributed environments and the ad-hoc networking of dynamic components is becoming important in all areas of applications. The continuing technical progress in computing and communication lead to an all-encompassing use of networks and computing power called ubiquitous or pervasive computing [1]. Pervasive computing system targets at constantly adapting their behavior in order to meet the needs of users within every changing physical, social, computing and communication context. Pervasive devices make ad-hoc connections among them and may be connected to different types of sensors to capture changes in the environment. In the evolution chain from centralized computing to pervasive computing as presented by [2] [3], Context awareness is at the heart of pervasive computing problems.

Context can be defined as an operational term whose definition depends on the intention for which it is collected and the interpretation of the operations involved on an entity at a particular time and space rather than the inherent characteristics of the entities and the operations themselves according to Dey & Winogard [4, 5]. The complexity of such problems increases in multiplicative fashion rather than additive with the addition of new components into the chain.

Pervasive context aware systems must address many of the requirements of traditional distributed systems such as heterogeneity, mobility, scalability and tolerance for component failures and disconnections.

In this paper, Section 2 discuss about related works in the field of context modeling. Section 3 with drawbacks of ontology based context model. Section 4 discuss about pros and cons of object relational database. Section 5 describes the proposed work.

2 Related Works

Sensors are used to get data from applications that are required for modeling. Sensors can be physical, virtual or logical sensors [6]. Physical data are captured using physical sensors. Virtual sensors are software processes which refer to data context. Logical sensors are the hybrid of the physical and virtual sensors and are used to solve complex tasks. After collecting the data from the application, it has to be represented in a suitable way for processing. Various modeling approaches are introduced to support standardization of techniques to present context for productive reasoning in different application area.

2.1 Classification of Context Modeling

The major classifications of context management modeling approaches are key-Value-Pair modeling, Graphical modeling, object oriented modeling, logic based modeling, Markup scheme modeling and Ontology modeling [3].

Key-Value –Pair modeling is the simplest category of the models. They are not very efficient for sophisticated and structuring purposes. It supports only exact matching and no inheritance. For example to set room no.109 as the location the command is “set LOC=ROOM=109”. The simplicity of key-value pair is a drawback if quality meta-information or ambiguity is considered. It is weak on the requirements. Graphical modeling is particularly useful for structuring. It is not used on instance level. The context is represented using Unified Modeling Language (UML) a firm graphical component and Contextual Extended Object Role Modeling (ORM). Because of its descriptive structure, UML is appropriate to model the context. Basic modeling concept in ORM is the fact. The modeling of a domain involves indentifying proper fact types and roles. Extended ORM is allowed to categorize fact types either as static or as dynamic. Structuring is done using this, but not extended on instance level. This method is mainly used to describe the structure of contextual knowledge and derive code or an Entity Relation model from model.

Logic based modeling uses logic expressions to define conditions on which a concluding expression or fact may be derived from a set of other expressions or facts.

Context is defined as facts, expressions and rules and has a high degree of formality. Object oriented modeling has a strong encapsulation and reusability feature. A high level of formality is reached because of well defined interfaces. Encapsulation is a drawback as the contexts are invisible.

Markup languages are a standard encoding system that consists of a set of symbols inserted in a document to control its structure, formatting, or the relationship among its parts. Markup languages establish the “vocabulary”, “grammar” and “syntax” of the codes applicable to text, image, or other form of data within an electronic document. The most widely used markup languages are SGML, HTML and XML. HTML is used for rendering the document, and XML is used for identifying the content of the document. Ontology is defined as explicit specification of a shared conceptualization [5]. Concepts and facts are modeled as context. It is firm in the field of normalization and formality.

2.2 Requirement of Context Modeling

Ubiquitous computing systems make high demands on context modeling approach in terms of the Distributed composition (dc), Partial validation (pv), Richness and quality of information (qua), Incompleteness and ambiguity (inc), Level of formality (for), Applicability to existing environments (app) [3]. Context model and its data varies with notably high dynamics in terms of time, network topology and source (dc). Partial validation (pv) is highly desirable to be able to partially validate contextual knowledge on structure as well as on instance level against a context model in use even if there is no single place or point in time where the contextual knowledge is available on one node as a result of distributed composition. This is important because of complexity of contextual inter relationships, which make any modeling intention error-prone. The quality of information delivered by a sensor varies over time as well as the richness of information provided by different kinds of sensors. Context model must support quality (qua) and richness indications. Contextual information may be incomplete, if information is gathered from sensor networks. Context model must handle interpolation of incomplete data on the instance level (inc). Contextual facts and interrelationships must be defined in a precise and traceable manner (for). A context model must be applicable within the existing infrastructure of ubiquitous computing environments (app).

Based on a survey made on systems from each category of context modeling approaches using the above requirements as a reference, [3] summarizes the appropriateness of modeling approaches for pervasive computing and among all the modeling approaches, ontology modeling category is more suitable for context aware computing. Context representation and management are introduced as a key feature in context-aware computing.

2.3 Ontology Based Context Modeling

The term is obtained from philosophy, where ontology is a systematic account of Existence. Context is modeled as concepts and facts using ontology. Some context aware systems that use these approaches are discussed below. CONtext ONtology stands for CONON [7]. It supports interoperability of different devices. The upper

ontology holds general concepts which are common to the sub domains and can therefore be extended. This is an infrastructure based environment. CoBrA-ONT [8] is a context management model that enables distributed agents to control the access to their personal information in context-aware environments. The center of this architecture is context broker agent. It depends on the assumption that there always exists a context-broker server that is known by all the participants. CoBrA is infrastructure-centric and is not for pervasive computing. SOUPA (Standard Ontology for Ubiquitous and Pervasive Applications) [9] is expressed in OWL and represents intelligent agents with associated beliefs, desires and intension, time etc for security and privacy. GAS ontology [10] is ontology designed for collaboration among ubiquitous computing devices. It provides a common language to communication and collaboration among the heterogeneous devices that constitute these environments. It supports the service discovery mechanism.

3 Restrictions of Ontology Context Management

Context aware systems are based on ad-hoc models of context, which causes lack of the desired formality and expressiveness. Existing models do not separate processing of context semantics from processing and representation of context data and structure. Ontology representation tools are suitable for statically representing the knowledge in domain. They are not designed for capturing and processing constantly changing information in dynamic environment in a scalable manner. Existing ontology languages and serialization formats are text based (xml/rdf/owl) and not designed for efficient query optimization, processing and retrieval of large context data. The main drawbacks of pure ontological approaches are low performance and data throughput.

4 Comparison of Relational Database(rdbms) and Object Relational Database (ordbms)

Relational models provide standard interfaces and query optimization tools for managing large and distributed context database or receive and send notification on context changes. Relational models are not designed for semantic interpretation of data.

Table 1. Comparison of Relational Database and Object Relational Database(ordbms)

Necessary Feature	Relational Approach	Ontology Approach	Object Relational Approach
Semantic Support	No	Yes	No
Ease of transaction (large data)	Yes	No	Yes
Query optimization	Yes	No	Yes
Reasoning support	No	Yes	No
Formality	Yes	Yes	Yes
Scalability	Yes	Yes	Yes

Relational database alone cannot be used to represent context in a pervasive environment. For semantic interpretations, ontology is used along with relational database. The table 1 shows the appropriateness of both approaches in relation to the necessary features. Both approaches have strong and weak sides with respect to features for context management modeling. Best of both sides are combined.

From the above Table 1 both relational approach and object relational approach are in the same level. Ordbms is more suitable than Relational approach because of the following advantages: Object relational approach ensures large storage capacity, which is an important part in web based development. The access speed is fairly quick. Ordbms have a massive scalability compared to relational approach. It boasts excellent manipulation power of object databases. It supports rich data types by adding a new object-oriented layer. The systems are initially implemented by storing the inactive context to a relational database and active context to an ontology. Then the response time to get the relevant time is noted. Further system is implemented by replacing the storing of inactive context to rdbms by object relational database. Then appropriate service can be provided to the user using service discovery [12].

5 Architecture of Proposed Work

The block diagram of the proposed context aware system has three layers is shown in Fig. 1. They are layers are Context acquisition layer (gets information using sensors), context middleware(represents context data in hierarchical form and stores in object relational database) and application layer(appropriate context is sent to user). The steps given below are used to filter the valid context into to object relational data base[13,14].

- Step 1: If system is running state repeat step 2 to 8
- Step 2: Assume new Context to null
- Step 3: New Context Class is assigned to Context Block
- Step 4: Repeat step 3 until no new Context is available
- Step 5: Get new Context from Context Block
- Step 6: Assign reliability Factor=0
- Step 7: If Context Block has Reliable Source then assign reliability Factor to 1 Else estimate Source Reliability
- Step 8: if reliability Factor is greater than Context Block reliability Threshold then Add Context to Object Relational Database else Context is in valid. Error message sent

Context selector uses historic and current user information, devices availability, institutional policies etc. to select and load only part of context to reasoning space.

Rules come from three different sources. They are rules defined by user, rules derived from organizational policies and rules derived from history data of past decisions using rule mining.

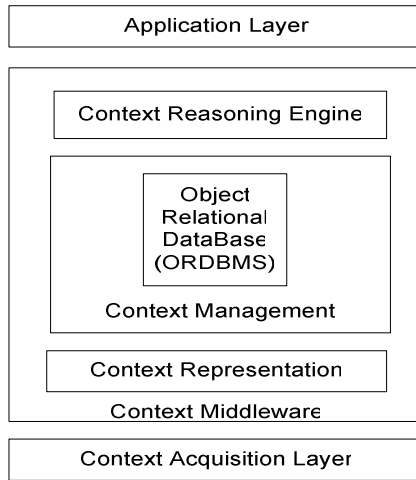


Fig. 1. Block Diagram of proposed system

5.1 Context Representation

Context can be represented as Entity, Hierarchy, Relation, Axiom and Metadata [11]. Set of entities for which context is to be captured is called as Entity. Set of binary relations that form an inversely directed acyclic graph (inverted DAG) on entities is hierarchies. *Nodes* of the graph represent entities and *arcs* of the graph represent hierarchical relations.

The root entity at the top of the hierarchy graph is a global entity known as *ContextEntity*. Union of the sets of binary relations R_c and R_a stands for Relation. R_c is a set of binary relations having both its domain and range from the set of entity. R_a is set of binary relations defined from set of entities to set of literals representing entity attributes. Domain of the relation R_a is the set of entity while its range is set of literal values. Axiomatic relation (A) is a relation about relation. Relation can be generic or domain based. Relation can be transitive, symmetry or inverse property. Meta data (M) is a data about data. It is important in context modeling to associate quality, precision, source, time stamps and other information to the context data. Such information is important to prepare the context data for reasoning and decisions. In hierarchy representation metadata information is a relation that describes another relation instance. Hierarchy is an important structure to organize and classify context entities and relations into two layers. They are generic layer or domain layer. Layered organization is used to classify and tag context data as generic domain independent or as domain dependent. For example Person isEngagedIn Activity (generic level), corresponding domain level is Physician isEngagedIn Patient treatment.

5.2 Mapping Hierarchical Context Representation to Relational Database

A step-by-step mapping algorithm from the hierarchy components to relational schema is given as follows:

- Step 1 : Collect all context entities in the hierarchy model and create table with Attributes Entity table (Context Entity, direct hierarchy relation, Layer)
- Step 2 : Collect all non hierarchical relations (other than isa and isInstanceOf) in Hierarchy and create a table with attributes (Relation, Persistence)
- Step 3 : Collect all relation instances in the hierarchy model and create a table with Attributes (Relation name, Context Entity, Value)
- Step 4 : Collect all context instances in the hierarchy model and create a table with Attribute (Entity Instance, Context Entity)
- Step 5 : Collect all relation defined on instances in the hierarchy model and creates A table with attribute (Entity Instance, Relation, Value, Time Stamp, Context Source and Context Precision)

5.3 Mapping Hierarchical Conceptual Model to UML

Unified Modeling Language (UML) is used to formalize hierarchical as a conceptual context representation model. UML is a standard specification language for object modeling. UML is used as a tool for ordbms designing. The Mapping of Hierarchical and UML is shown in table 2.

Table 2. Mapping Hierarchical and UML

Hierarchical	UML
Entity (E)	UML Class
Hierarchy Relation (H)	Generalization Relationship
Entity Relation (R_e)	Association Relationship
Attribute Relation (R_a)	Attributes in UML Class
Axioms (A)	Association Classes
Metadata (M)	Association Classes

5.4 Need for Semantics

Consider the situation of staff members' (Ben , Dan and Rita) tea break scenario in where Ben isLocatedIn Room-305 at time 2010022310, Dan isLocatedIn Room-305 at time 2010022310 and Rita isLocatedIn Office-305 at time 2010022310. A simple query (select Subject from table.context where predicate="isLocatedIn" and Object="Room-305") select "Ben" as an output. By common sense, terms "Office" and "Room" are synonymous in the domain of interest, the output must be "Ben" and "Rita" to the query. This example demonstrates the need for a context model that describes concepts, concepts hierarchies and their relationships. The concepts of *office* and *Room* in the above tea break scenario are same and using OWL language and *owl:sameAs* property the problem can be solved.

5.5 Mapping Hierarchical Conceptual Model to Ontology

Ontology provides standardization of the structure of the context representation, semantic description and relationship of entities. For example, to define the similarity axiom between the concepts *ownerOf* and *owns*, *owl:sameAs* property is used. Similarly, the

Table 3. Mapping hierarchical model and ontology

Hierarchical Model	Ontology
Entity (E)	Owl:class
Hierarchy Relation (R)	rdfs:subClassOf, rdfs:superClassOf, rdf:type
Entity Relation (R _e)	Owl:objectproperty
Entity Attribute (R _a)	Owl:dataTypeProperty
Axioms (A)	properties (owl:TransitiveProperty, owl:inverseOf, ...) restrictions(hasValue, hasMinCardinality,someOf, ...)
Metadata (M)	rdf reification

symmetric axiom on the concept of *coLocatedWith* can be defined using *owl:symmentricProperty* and the inverse relationship property between *ownerOf* and *OwnedBy* can be defined using *owl:inverseOf Property*. Mapping between hierarchical model and ontology is shown in Table 3.

5.6 Heuristic Selection of Context

A reasoning space is a search space from which the right set of information is extracted to perform reasoning and inferences. A search space is a set of all possible solution to a problem. Uninformed search algorithms use the intuitive method of searching through the search space, whereas informed search algorithms use heuristic functions to apply knowledge about the structure of the search space to try to reduce the amount of time spent on searching. The entire set of context data is virtually organized into the hierarchical graph. The graph consists of hierarchical tree of context entities and their corresponding relations, axioms and metadata. Pruning techniques on the hierarchical tree of context entities in the graph is used to minimize the size of the reasoning space[14]. The possible input and output is given in table 4.

Table 4. Input and Output of heuristic method

Input	Output
T set of all entities in the Entity Relation hierarchy graph	S set of relevant entities selected from the set T
E ₁ set of entities identified at the time of initialization	
E ₂ set of entities specified at the time of initialization	S is a subset of T.
getProbability (function to predicate the probability of occurrence based on history data)	
getTH() and estimateTH() (functions that set threshold value)	

Context data are loaded into the reasoning space selectively by pruning the irrelevant part of the data. To improve both quality and speed , the selection (pruning) must be selected such a way that the values are nearing zero. In real time , it means that the performed selection/pruning matches with the users’ intension. This on other hand depends on the prediction module and how reliable the history data or the

experience is. It is like human beings where, under normal condition, experiences improve performances both in terms of quality and speed. The context model using pure ontology and object relational database is implemented.

5.7 Performance Issues in Selection Process

Context entities are parameters from which the contents of the reasoning space (context data) are defined. Two measures of performance of the algorithm are accuracy (quality) of reasoning and the response time (speed). Context data are loaded into the reasoning space selectively by pruning the irrelevant part of the data. The context model using pure ontology and using ontology and object relational database is implemented. The x axis has data size ranging from 200 to 9,000. The x axis is in the scale is 10 unit is equal to 1000 .y-axis has the average time in milliseconds (ms).Figure 2 shows the response time by using pure ontology approach. From Fig 2. It is clearly shows that ordbms approach is better than ontology approach.

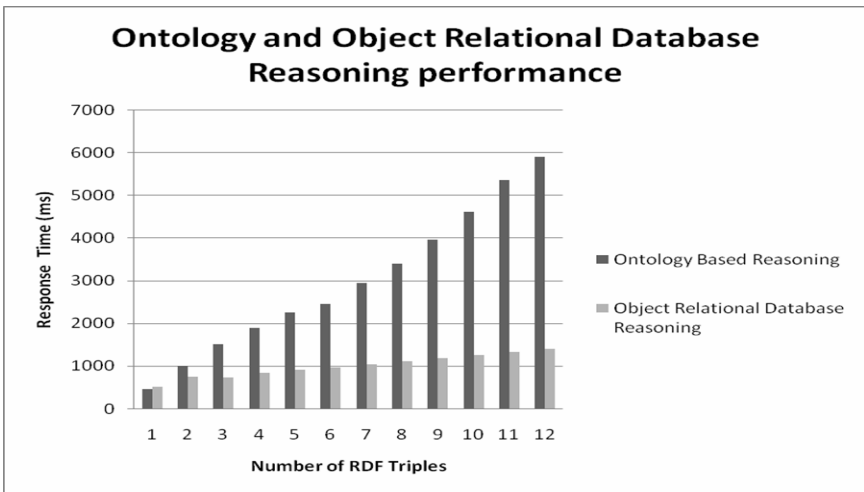


Fig. 2. Comparison of ontology and Object Relational Database Approach

6 Conclusion

A layered representation of context using layered and directed graph is proposed. It helps to classify and tag context data as generic domain independent or as domain dependent. A context model using object relational database is proposed. It is compared with a context model using pure ontology. It has been proved that object relational database approach is better than pure ontology based model with respect to response time. This paper focuses on context representation, storage of context and pruning technique. The future work includes reasoning, decision making of the context obtained from the context management.

References

- [1] Mattern, F., Sturn, P.: From Distributed Systems to Ubiquitous Computing- State of the Art. In: Trends and Prospects of Future Networked system, Fachtagung Kommunikation in Verteilten Systemen (KiVS), Leipzig, pp. 3–25. Springer, Berlin (2003)
- [2] Satyanarayanan, M.: Pervasive Computing: Vision and Challenges. *IEEE Personal communication*, 10–17 (August 2000)
- [3] Strang, T., Linnho -Popien, C.: A Context Modeling Survey. In: The Proceedings of the First International Workshop on Advanced Context Modeling, Reasoning and Management, Sixth International Conference on UbiComp 2004, Nottingham, England (2004)
- [4] Winograd, T.: Architectures for Context. *Human-Computer Interaction* 16(2, 3, 4), 401–419 (2001)
- [5] Dey, A.K., Abowd, G.D.: Towards a Better Understanding of Context and Context Awareness. In: Proceedings of the CHI Workshop on the What, The Hague, the Netherlands (April 2000)
- [6] Baldauf, M.: A survey on context aware Systems. *Int. J. Ad hoc and Ubiquitous Computing* 2(4), 263–277 (2007)
- [7] Wang, X., Zhang, D., Gu, T., Pung, H.K.: Ontology Based Context Modeling and Reasoning using OWL, workshop on context modeling and reasoning. In: IEEE International Conference on Pervasive Computing and Communication, Orlando, Florida (March 2004)
- [8] Chen, H.: An Intelligent Broker Architecture for Pervasive Context-Aware Systems. PhD Thesis University of Maryland, Baltimore County, USA (December 2004)
- [9] Chen, H., Perich, F., Finin, T., et al.: SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications. In: International Conference on Mobile and Ubiquitous Systems: Networking and Services, Boston, USA (2004)
- [10] Christopoulou, E., Kameas, A.: Gas Ontology: ontology for collaboration among ubiquitous computing devices ubiquitous computing devices. *International Journal of Human-Computer Studies* 62(5), 664–685 (2005)
- [11] Ejigu, D., Scuturi, M., Burnia, L.: An Ontology Based Approach to Context Modeling and Reasoning in Pervasive Computing, percom. In: Fifth IEEE International Conference on Pervasive Computing and Communications Workshops (PerComW 2007), pp. 14–19 (2007)
- [12] Vanathi, B., Rhymend Uthariaraj, V.: Ontology based service discovery for context aware computing. In: First International Conference on Advanced Computing, ICAC 2009 (2009)
- [13] Vanathi, B., Rhymend Uthariaraj, V.: Context Representation and Management in a Pervasive Management. In: Das, V.V., Vijaykumar, R. (eds.) ICT 2010. Communications in Computer and Information Science CCIS, vol. 101, pp. 543–548. Springer, Heidelberg (2010)
- [14] Vanathi, B., Rhymend Uthariaraj, V.: Context Representation Using Hierarchical Method and Heuristic Based Context Selection in Context Aware Computing. In: Proceedings of the ACM-W 1st Conference of Women in Computing in India: Emerging Trends in Computer proceedings

An Analytical Model for Sparse and Dense Vehicular Ad hoc Networks

Sara Najafzadeh, Norafida Ithnin, and Ramin Karimi

Faculty of Computer Science and Information System
University Technology Malaysia
{sara60,nithnin,rakarimi1}@gmail.com

Abstract. Vehicular Ad hoc Network (VANET), a subclass of mobile ad hoc networks (MANETs), is a promising approach for the intelligent transportation system (ITS). The design of routing protocols in VANETs is important and necessary issue for support the smart ITS. The key difference of VANET and MANET is the special mobility pattern and rapidly changeable topology. Besides, the temporary network fragmentation problem and the broadcast storm problem are further considered for designing protocols in VANETs. The temporary network fragmentation problem which happens in sparse network caused by rapidly changeable topology influence on the performance of data transmissions. The broadcast storm problem in dense network seriously affects the successful rate of message delivery in VANETs. This paper presents an analytical model for sparse and dense Vehicular Ad hoc Networks.

Keywords: Vehicular Ad hoc Network, Routing, Broadcasting, Traffic density, Disconnected Network.

1 Introduction

Vehicular Ad hoc Networks (VANET), a new technology to build a wireless network between vehicles (V2V). VANETs are based on short-range wireless communication (e.g., IEEE 802.11) between vehicles. The Federal Communication Commission (FCC) has allocated 75 MHz in 5.9 GHz band for Dedicated Short Range Communication (DSRC). DSRC was conceived to provide architecture for vehicles in Vehicular Network to communicate with each other and with infrastructure. In DSRC, subsequently specialized as Wireless Access in Vehicular Environment (WAVE), GPS-enabled vehicles that are equipped on-board units can communicate with each other.

VANETs are special class of Mobile Ad hoc Networks (MANETs). The major characteristics as compared to MANETs are following: components building the network are vehicles, dynamic topology, geographically constrained topology, vehicle mobility, frequently disconnected network and time-varying vehicle density [1]. VANETs could be playing an important role in future of vehicle communications. VANETs provide a variety of interesting applications like cooperative forward collisions, sharing emergency warning messages, weather and traffic data among vehicles.

Routing is one of the key research issues in vehicular networks as long as many applications require multi hop communications among vehicles. VANET routing

protocols are designed to deliver data packets to their intended destinations using available vehicles as relays. Vehicular networks exhibit unique properties such as a possibly large number of vehicles, high dynamics, and fast-changing vehicle densities and partitions. Even if a large population of vehicles is present, constrained mobility, traffic lights, intersections and the like lead to frequent network partitions and an uneven network density. These properties represent a real challenge for routing protocol designers.[2]

On the other hand, some characteristics of VANETs provide excellent support for routing protocols. For instance, mobility constraints, predictable mobility, and access to additional information (e.g., geographic coordinates, city maps, etc.) can be exploited by VANET-specific routing solutions to increase their performance. Routing messages in vehicular networks remain a great challenge. The applications in such networks have specific requirements. This means that VANET routing protocol designers are required to explore a distinctive set of technical challenges and design alternatives.

2 Research Background

The past few years have seen increasing technological advances in Vehicular Ad hoc Network. There are some significant challenges in VANETs for broadcast protocol.

First, high mobility and dynamic topology of vehicles is one of the issues should be consider in VANETs. In order to high speed of vehicles, topology of VANET is sometimes changing. Due to the same reason the connection between nodes frequently could be changed rapidly and unpredictably. So there is also a high degree of change in the number and distribution of the node in the network. Since vehicle mobility depends heavily on environment scenarios, the movement direction sometimes is predictable.

Second, high network density related to traffic density in Vehicular Network is the other challenge. Traffic density is high during peak hour in urban areas. Because of large number of vehicles , blindly flooding the packets which each node re-broadcast messages to its entire neighbor except the one it got the message from may lead to serious contention and collisions in transmission among neighboring nodes. This problem is mostly referred to as broadcast storm problem.

Third, disconnection is another challenging issue in designing a routing protocol in Vehicular ad hoc network. Apart from the studies conducted in dense networks, there is a growing need to deal with disconnection in sparse Vehicular Ad hoc Networks. Where nodes are distributed sparsely, most of the time path between source and destinations may not exist. Traditional routing protocol can only transmit packet over end-to-end connected path so they would fail in such conditions. Although it is very hard to find end-to-end connection for a sparsely networks, the high mobility of vehicular networks introduces opportunities for mobile vehicles to connect each other during the movements.

The broadcast storm problem has been heavily discussed in MANETs. [8] The broadcast storm problem causes serious packet collision and packet loss since too many vehicles simultaneously broadcast messages. A shortest routing path can be discovered by broadcasting the RREQ (route request) message. The broadcast routing

strategies developed are 1-persistence and p-persistence. [7] However, these two strategies easily cause the broadcast storm problem, especially in a high density network. Three distributed broadcast suppression techniques, weighted p-persistence, slotted 1-persistence, and slotted p-persistence schemes has been proposed.[7] In the weighted p-persistence scheme, if vehicle V_j receives a packet form vehicle V_i , vehicle V_j first checks whether the packet has been received. If vehicle V_j receives this packet at the first time, then vehicle V_j has probability p_{ij} to re-broadcast the packet. Otherwise, vehicle V_j drops this packet, under $P_{ij} = \frac{D_{ij}}{R}$ where D_{ij} is the distance between vehicle V_i and V_j , R is the transmission range. Neighbors of vehicle V_i change p_{ij} to 1 to ensure that the message must be broadcasted if they have no received the rebroadcast message after waiting a random time. In the slotted 1-persistence scheme, If vehicle V_j firstly receives this packet from vehicle V_i , then vehicle V_j waits for TS_{ij} time slots, vehicle V_j has probability 1 to re-broadcast the packet, where $TS_{ij} = ij \times \tau$.

Where τ is the propagation time for one hop transmission and N_s is the default number of time-slot.

$$S_{ij} = \begin{cases} \left\lfloor N_s \left(1 - \frac{D_{ij}}{R} \right) \right\rfloor, & \text{if } D_{ij} < R \\ 0, & \text{if } D_{ij} > R \end{cases} \quad (1)$$

The slotted p-persistence scheme combines the weighted p-persistence and slotted 1-persistence schemes. If vehicle V_j firstly receives the packet from V_i , then vehicle V_j waits for TS_{ij} time-slots. Vehicle V_j has probability p_{ij} to re-broadcast the packet. Simulation results show that these three schemes can achieve up to 90% reduction of packet loss rate.

DV-CAST for a multi-hop broadcast routing protocol in VANETs [8]. This paper indicates three traffic scenarios for a vehicular broadcasting; (1) dense traffic scenario, (2) sparse traffic scenario, and (3) regular traffic scenario. This paper integrates previously proposed routing solution to develop DV-CAST which is suitable for both of dense and sparse traffic scenarios. The overhead in dense traffic scenario can be reduced. DV-CAST utilizes the information form one-hop neighbors to make the routing decision. Each vehicle records the states of neighboring vehicles all the time. Three parameters should be recorded; (1) DFlag (Destination Flag), (2) MDC (Message Direction Connectivity), and (3) ODC (Opposite Direction Connectivity). The DFlag indicates whether the direction of the current vehicle is the same as the source vehicle. The MDC describes whether a vehicle exists behind the current vehicle to assist broadcast message forwarding. The ODC indicates whether a vehicle exists in the opposite direction. Vehicles can make decision based on these three parameters. Observe that, the temporary network fragmentation problem is also considered in the design of broadcasting methods for inter-vehicle communications system to provide emergency information dissemination in VANETs. The purpose of emergency information is to announce an urgent event by broadcasting for surrounding vehicles. According to the purposes of emergency information, the proposed broadcast methods

are divided into two categories, emergency-vehicle-approach information and traffic accident information [3]. Emergency-vehicle-approach information is used to announce the urgent event to those vehicles in front of the current vehicle, so the emergency information is only disseminated ahead. Traffic accident information is used to announce the urgent event to those vehicles behind the current vehicle; the emergency information is only disseminated behind. By limiting the broadcast direction, the proposed broadcast methods can provide broadcasts to a particular area and avoid mistakenly notifying other areas where the information is not needed [3]. To decide emergency information should be received, forwarded, or dropped, three conditions below are examined:

$$(\vec{Xc} - \vec{Xe} + \alpha \vec{Vf}) \cdot \vec{Vf} > 0 \tag{2}$$

$$|\vec{Xc} - \vec{Xe} + \alpha \vec{Vf}| < Rnotification \tag{3}$$

$$|\vec{Xc} - \vec{Xe} + \alpha \vec{Vf}| < Rrelay \tag{4}$$

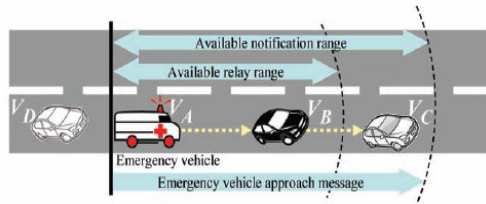


Fig. 1. The broadcast area for emergency information

Where XC is the location vector of the current vehicle, XE is the location vector of the emergency vehicle, VF is the forward direction vector of the emergency information, $Rnotification$ is the available notification range, and $Rrelay$ is the available relay range.

If conditions (2) and (3) are true, the current vehicle receives the emergency information. If conditions (2) and (4) are true, the current vehicle re-broadcasts the emergency information. Figure 1 shows that vehicle VA broadcasts the emergency message to the restricted direction. Vehicle VD does nothing. Vehicle VD is located in the relay range, it re-broadcasts the emergency information. Vehicle VD is located in notification range but not in relay range, VC just receives the emergency information and not to re-broadcast.

3 Sparse Vehicular Networks

One of the most important limitations of V2V networks is end to end delay due to partitioning of networks. It is observed that carry-and-forward is the new and key consideration for designing all routing protocols in VANETs. With the consideration of multi-hop forwarding and carry and-forward techniques, min-delay and delay-bounded routing protocols for VANETs are discussed in VANETs.

In this section, a data propagation model has been employed. In this section vehicles categorized in two groups.

1) Group 1:

This group refers to the nodes which carries the information along the road.

2) Group 2:

This group refers to the nodes which have not received information.

A distributed vehicle traffic model is used with the following assumptions:

- Poisson Arrival: Traffic moves on the road follows a Poisson process with an average rate equal to traffic flow rate(vehicles/time)
- Independent vehicle mobility: vehicle travels with a speed that follows a random distribution between [Vmin,Vmax].

By the assumption of undisturbed vehicle traffic model, vehicles passing location X follows a Poisson process [5].

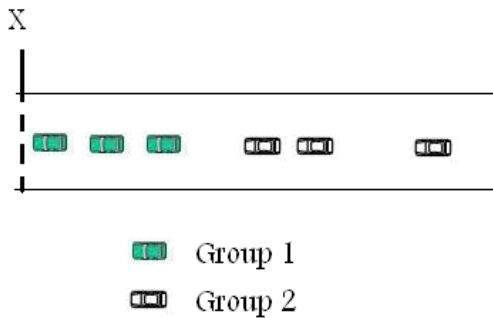


Fig. 2. Scenario

Assume n vehicles pass location X (Figure 2) during (0, t]. This leads to

$$T_i \sim \text{uniform}(0, t] \tag{5}$$

$$P[X(t) < X | N(t) = n] = P[V^* t < x] * P[V^*(t - T) < X]^n \tag{6}$$

Where T and T_i have same distribution and V and V_i have same distribution. Based on the assumption of Poisson process N(t) has Poisson distribution

$$f_{X(t)}(x) = \sum_{n=0}^{\infty} P[X(t) < x | N(t) = n] P[N(t) = n] \tag{7}$$

Then we have

$$P[X(t) < X | N(t) = n] = P[V_0^* x, V_i^*(t - T_i) < x \text{ for each } i=1, 2, \dots, n] \tag{8}$$

By the assumption of undistributed node traffic model, vehicle passing X follow Poisson process. Under this situation that n vehicles pass X during (0, t], the times at which these vehicles pass location X consider unordered random variables distributed both independently and uniformly in the interval (0,t], this leads to

$$T_i \sim \text{uniform}(0, t) \tag{9}$$

Since T_1, T_2, \dots, T_n and V_0, V_1, \dots, V_n are

$$P[X(t) < x | N(t) = n] = P[V^*_{t-x}] * P[V^*(t-T) < x]^n \tag{10}$$

Where T has the same distribution as T_i and V has the same distribution as V_i . Based on the assumption of Poisson process, $N(t)$ has the Poisson distribution

$$P[N(t) = n] = \frac{e^{-\lambda t}}{n!} \tag{11}$$

Let $V(t)$ be the message speed

$$E[V(t)] = \frac{\phi[E[X(t)]]}{\phi t} \tag{12}$$

$E[V(t)]$, the average message speed is given in the above equation. (Figure 3)

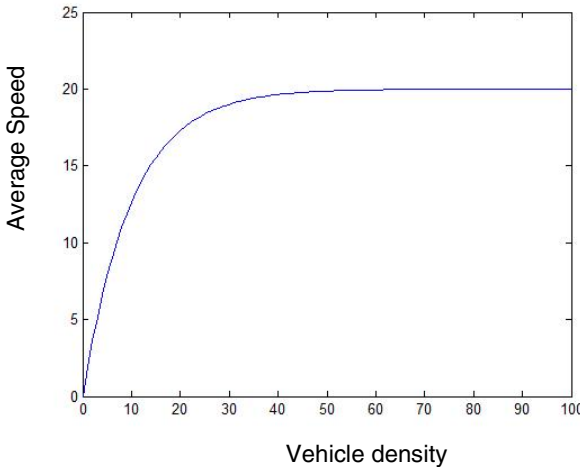


Fig. 3. Average message speed

4 Dense Network Model

Besides, the temporary network fragmentation problem and the broadcast storm problem are further considered for designing routing protocols in VANETs. The temporary network fragmentation problem caused by rapidly changeable topology influence on the performance of data transmissions. The broadcast storm problem seriously affects the successful rate of message delivery in VANETs [4].

In this section a dense network will be modeled. For a given time t , we define $Vp(t)$ to be the average message propagation speed during the interval $[0, t]$, $p=X(t)/t$. we define the long-term average message propagation speed vp to be

$$Vp = \lim_{t \rightarrow \infty} V(t) = \lim_{t \rightarrow \infty} x/t \tag{13}$$

So

$$V_{avg} = \frac{E[T1]V + E[T2]V_{radio}}{E[T1] + E[T2]} \tag{14}$$

Where $T1$ is the average time traveled during forward phase and $T2$ is the time spent during forward phase, V is Vehicle speed and V_{radio} indicate multihop radio propagation speed. Figure 4 indicate numerical result related to average speed.

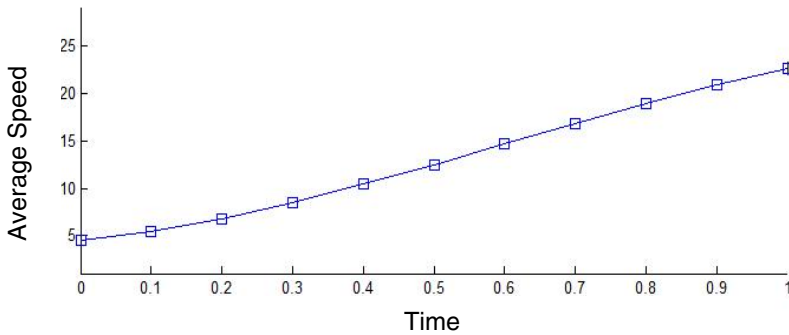


Fig. 4. Numerical result

5 Conclusions

Besides, the temporary network fragmentation problem and the broadcast storm problem are further considered for designing routing protocols in VANETs. The temporary network fragmentation problem caused by rapidly changeable topology influence on the performance of data transmissions. The broadcast storm problem seriously affects the successful rate of message delivery in VANETs. To understand the characteristics of VANETs in different network densities We presented models for one-way vehicle traffic where information is propagating in the direction of vehicle traffic flow. Then we develop a collection of models to study information propagation using V2V communications. These models can provide expected information propagation speed/distance over time.

References

- Blum, J., Eskandarian, A., Hoffman, L.: Challenges of inter-vehicle ad hoc networks. IEEE Transactions on Intelligent Transportation Systems 5(4), 347–351 (2006), doi:10.1109/TITS.2004.838218

2. Chen, Y.S., Lin, Y.W., Lee, S.L.: A Mobicast Routing Protocol for Vehicular Ad Hoc Networks. In: *Mobile Networks and Applications (MONET)*, ACM/Springer, New York (2009)
3. Fukuhara, T., Warabino, T., Ohseki, T., Saito, K., Sugiyama, K., Nishida, T., Eguchi, K.: Broadcast Methods for Inter-vehicle Communications System. In: *IEEE Wireless Communications and Networking Conference*, vol. 4, pp. 2252–2257 (2005)
4. Hughes, L., Maghsoudlou, A.: An Efficient Coverage-Based Flooding Scheme for Geocasting in Mobile Ad Hoc Networks. In: *International Conference on Advanced Information Networking and Applications*, vol. 1, (2006)
5. Naumov, V., Baumann, R., Gross, T.: An Evaluation Of Inter-Vehicle Ad Hoc Networks Based On Realistic Vehicular Traces. In: *ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 108–119 (2006)
6. Tonguz, O., Wisitpongphan, N., Bai, F., Mudalige, P., Sadekar, V.: Broadcasting in VANET. *Mobile Networking for Vehicular Environments*, 7–12 (2007)
7. Tonguz, O.K., Wisitpongphan, N., Parikh, J.S., Bai, F., Mudalige, P., Sadekar, V.K.: On the Broadcast Storm Problem in Ad hoc Wireless Networks. In: *International Conference on Broadband Communications, Networks and Systems*, pp. 1–11 (2006)
8. Tseng, Y.C., Ni, S.Y., Chen, Y.S., Sheu, J.P.: The Broadcast Storm Problem in a Mobile Ad Hoc Network. [WINET]. *ACM Wireless Networks* 8(2), 153–167 (2002), doi:10.1023/A:1013763825347

MIMO Ad Hoc Networks-Mutual Information and Channel Capacity

Chowdhuri Swati¹, Mondal Arun Kumar², and P.K. Baneerjee³

¹ Department of Electronics and Communication Engineering
Seacom Engineering College, Howrah, W.B., India
swati.chowdhuri@gmail.com

² Department of Electronics and Communication Engineering
Guru Nanak Institute of Technology, Kolkata, W.B., India
akmondal_31@yahoo.co.in

³ Department of Electronics and Telecommunication Engineering
Jadavpur University, Kolkata, W.B., India
pkbanj@rediffmail.com

Abstract. With the rapid growth of wireless communication infrastructure over the recent few years, new challenges has been posed on the system and analysis on wireless ad hoc networks. Ad hoc network functionalities and securities using the single path as well as multi paths routing are analyzed by many researchers [1,2,3] But, one of the most important events in wireless communication is the advent of MIMO technologies which uses multiple antennas to coherently resolve more information than possible using a single antenna [4,5]. In fact, both the Wi-Fi (IEEE 802.11n) standard and the WiMAX (IEEE 802.16e) standard incorporated MIMO transmitter and receiver by adding a 40MHz channels to the physical layer (PHY) and frame aggregation to the MAC layer[6]. The key advantage of MIMO wireless communication system lies in their ability to significantly increase the wireless channel capacity as well the data rates without requiring additional bandwidth. The MIMO channel model development originates from the fundamental of information theory [7, 8, 9]. In this paper we presented a MIMO Ad Hoc network channel model where with the received signal, interference and noise is added. Here the interference and noise is assumed to be Gaussian and optimal distribution of the transmitted signal is also symmetric complex Gaussian. The ergodic mutual information between the channel input and output is computed in terms of two conditional entropies of the channel and this is maximized over a set of transmitted signal to compute the channel capacity. Finally some simulated results are shown for the channel capacities with different set of input transmitting and output receiving antennas and SNR. And we get maximum spectral efficiency for a certain set of antennas. The simulated result shows that the there is no need to increase the number of antennas after certain limit, that means number of antenna becomes constant for maximize spectral efficiency.

Keywords: ad hoc networks, Multi Input Multi Output (MIMO), channel capacity, Signal to Noise ratio, mutual information.

1 Introduction

Wireless ad hoc networks is basically peer to peer network of mobile hosts that have no any communication infrastructure as well as base station. Wireless ad hoc network is a collection of nodes that can move freely and communicate with each other using wireless device in the fly. Each node has to act simultaneously as a host as well as a router. Nodes have to co-operate each other in the task of communicating packets to the destination node. For the nodes that are not within the direct communication range in the network work collectively to relay packets from them. This network is characterized by its dynamic topological changes, limited battery power of nodes. The network topology can change frequently and dynamically when a nodes moves into or out of the transmission range of another nodes, the link between the two nodes become up or down [1, 2]. The rapid changes of topology due to mobile hosts are distributed among the node that degrades the data rates [3]. The topological change causes the unstable wireless link which may up or down due to signal fading, interference from other signal or changing the transmission power. This degrades the data transmission rate through the channel. Nodes in this type of network use the Distributed Co-ordinate Function (DCF) medium access protocol such as IEEE802.11 to reserve local access to the wireless medium [1,6]. Multi Input Multi Output (MIMO) ad hoc network systems are the promising techniques for improving the data rates particularly in frequency selective fading environments. The IEEE802.11n medium access protocols are introduced to extend the concept of MIMO communication in mobile ad hoc network application. The wireless LANs that use IEEE802.11n standard provide much higher spectral efficiency such as high data throughput, improved system performance, etc than that of ordinary wireless system [3, 6].

Each node has to act simultaneously as a host as well as a router. Nodes have to co-operate each other in the task of communicating packets to the destination node. The rapid change of topology due to mobile hosts are distributed among the nodes that degrades the data rates . Multi Input Multi Output (MIMO) ad hoc network systems are the promising techniques for improving the data rates particularly in frequency selective fading environments. The availability of multiple antennas at the transmitter and receiver end in MIMO system improves the data rates by a factor of M times where $M = \min \{M_t, M_r\}$ and M_t and M_r are the number of antennas in the transmitter and receiver respectively.

Utilizing multiple transmitter and receivers communicate over a Multiple-Input-Multiple-Output(MIMO) channel can provide substantial increase in channel capacity. Using IEEE802.11n, maximization of the channel capacity can be made knowing system full channel state information available at the transmitting end. The multipath communication using Multi Input Multi Output (MIMO) mitigates the problem of decrease of data rate and consequently enhances the channel capacity [7, 8]. In this paper we present a MIMO ad hoc network and will discuss the mutual information and channel capacity of the MIMO ad hoc network. The availability of multiple

antenna at the transmitter and receiver end in MIMO system improves the data rate by a factor of M times where $M = \min\{M_t, M_r\}$ and M_t and M_r are the number of antennas in the transmitter and receiver respectively. We have computed the simulated results for the channel capacity vs cumulative distributed function (CDF) with different signal to noise ratio for the ordinary ad hoc network and MIMO ad hoc network system. Though the channel capacity increases with the number of antennas yet there is some limitation in the increase of channel capacity with the number of antennas for a particular SNR of the channel. In this paper we have simulated the model to find the maximum number of antennas for which the spectral efficiency is maximized.

2 MIMO Ad Hoc Network Model

We consider a MIMO system over a wireless channel with M transmitter antennas and N receiving antennas then the input-output relation can be written as

$$y(k) = HX(k) + \eta(k) \tag{1}$$

$$y(n) = \begin{bmatrix} y_1(nT_s) \\ y_2(nT_s) \\ \cdot \\ y_N(nT_s) \end{bmatrix}$$

$$\text{and } x(n) = \begin{bmatrix} x_1(nT_s) \\ x_2(nT_s) \\ \cdot \\ x_N(nT_s) \end{bmatrix}$$

and the element H is a M x N matrix that have a complex Gaussian distribution. $\eta(k)$ is the N*1 additive white Gaussian noise vector with zero mean and covariance C_w .

If the set of signals are independent and excited by the different transmitter and receiver to detect then the element H will be independent as well. It is observed that if these two conditions are satisfied maximum MIMO channel capacity will be obtained for a given transmitter power [10].

In MIMO ad hoc network each node with multiple antennas will behave as a host as well as a router. For an instant of time, let K number of transmitter with M number of transmitting antennas and receiver with N number of receiving antennas pair are

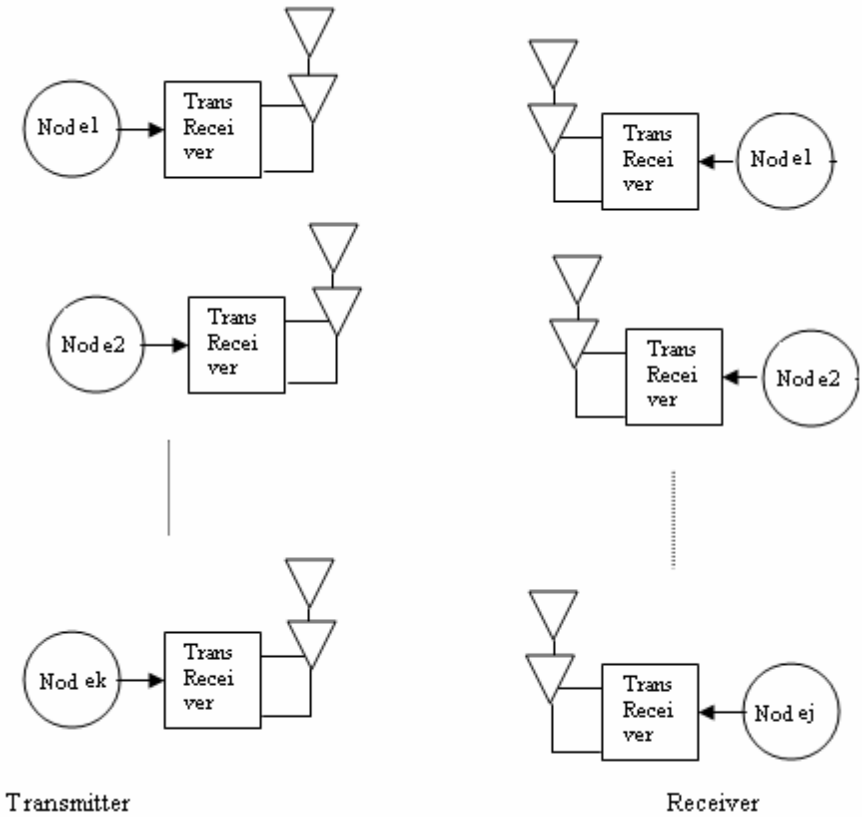


Fig. 1. MIMO Ad-Hoc network model

communicating. The channel matrix for the channel between the receiver antenna k and transmitter antenna j is denoted by $H_{kj} \in \mathbb{C}^{M \times N}$.

And for k of K links transmitted signal vector $x_k \in \mathbb{C}^{M \times 1}$ is independent across the antennas.

With the received signal the interference and noise is added. Let the interference plus noise is circularly complex Gaussian, the optimal distribution of the signal X_k is also circularly symmetric complex Gaussian. The channel is known to the receiver, then the ergodic mutual information between the channel input and output can be expressed in terms of two conditional entropies: $E(h(y_k))$ is known as the self entropy of the channel without any knowledge about the received signal y_k and $E(h(y_k / X_k))$ is known as the entropy of the channel with the knowledge of a particular received signal y_k .

The mutual information $I(X;Y)$ between the channel input and output is the amount of uncertainty of difference of the above two entropies. Thus,

$$\begin{aligned}
 I(X;Y) &= E((h(y_k)) - (h(y_k / X_k))) \\
 &= E(h(y_k) - h(n)) \\
 &= E(\log \det(I + \eta_k H_{kj} R_j H_k^H (\sum_{j \neq k} \eta_{kj} H_{kj} R_j H_{kj}^H + \sigma^2 I)))
 \end{aligned}$$

The channel capacity is the maximum mutual information between the input and output of the channel over a set of input signals. So, it can be written as

$$C_k = \max_{\{X_k\}} E(\log \det(I + \eta_k H_{kj} R_j H_k^H (\sum_{j \neq k} \eta_{kj} H_{kj} R_j H_{kj}^H + \sigma^2 I)))$$

The network spectral efficiency is defined as the sum of pair wise spectral efficiency between intended transmitter and receiver. The overall MIMO channel capacity is the sum of the channel capacity over the available links i.e.

$$C = \sum_{k=1}^K C_k$$

Here C_k indicates capacity of a point-to-point MIMO system and C gives the sum of the transits for $C_k = 1,2,..$ as the number of diagonal entries that are significantly larger than others. We use $\eta^{(dB)}$, the SNR of the channel and consider the weak interference level such that the $\eta^{(dB)}$ will be prominent to calculate the channel capacity. Here we consider the different SNR value of the channel to compute the channel capacity and cumulative distribution Function (CDF). In the subsequent section we have shown the simulated results.

3 Simulated Results

The capacity of a MIMO ad hoc network using different number of transmitting and receiving antennas are illustrated in figure2, 3 and 4.

If we increase the number of transmitting and receiving antenna for the ad hoc network it will give the high channel capacity than ordinary ad hoc network system. At the higher SNR value when channel noise or interference reduces it will give the higher value of channel capacity.

A comparative study of the Spectral Efficiency with different number of antennas shown in the table1. The simulated result for spectral efficiency vs number of antennas is shown in fig 5.

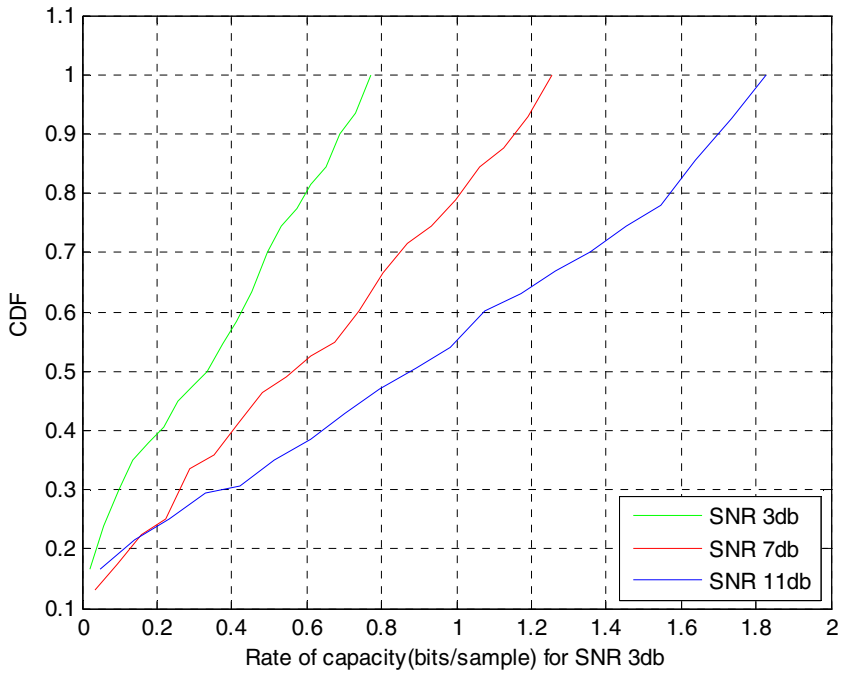


Fig. 2. Ergodic Channel Capacity Vs CDF Plot with Standard Ad Hoc Network at different SNR

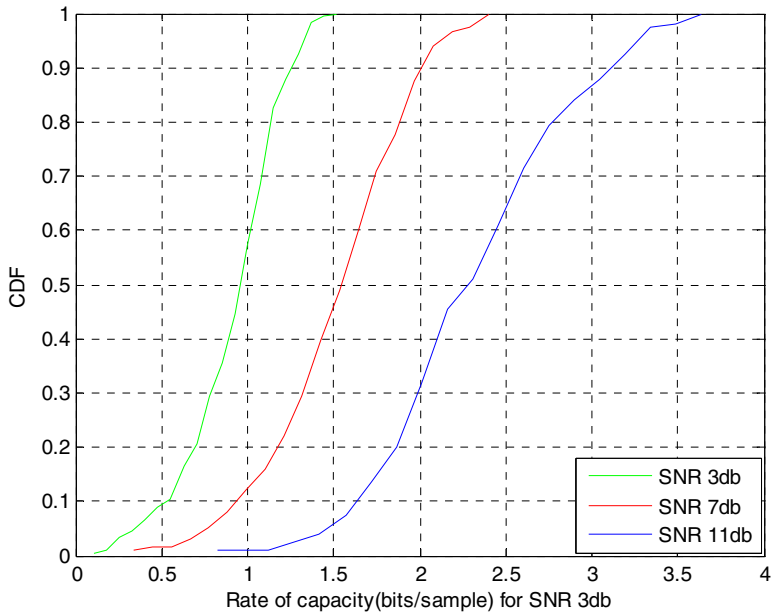


Fig. 3. Ergodic Channel Capacity Vs CDF Plot with 2X2 MIMO Ad Hoc Network at different SNR

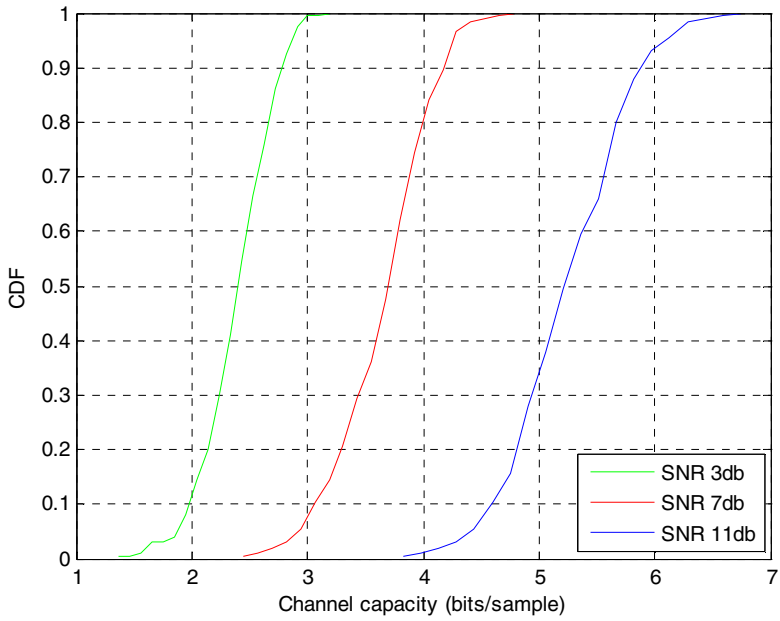


Fig. 4. Ergodic Channel Capacity Vs CDF Plot with 4X4 MIMO Ad Hoc Network at different SNR

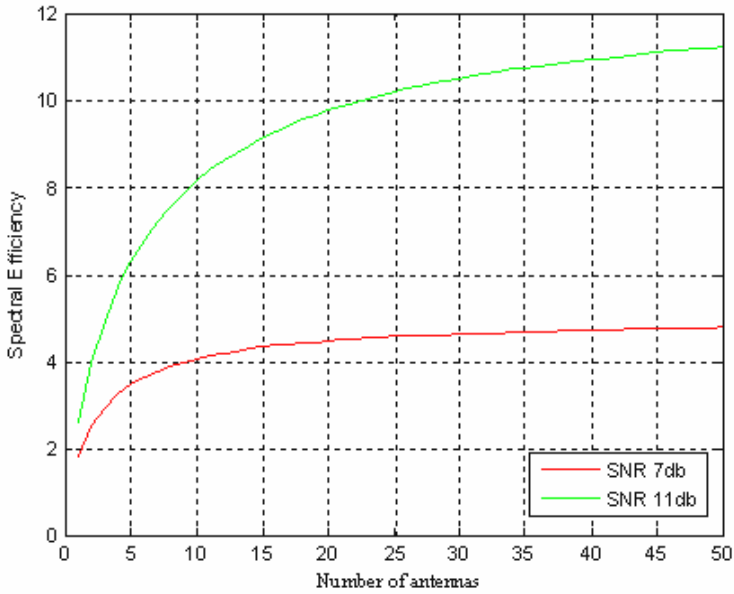


Fig. 5. Number of antenna Vs Spectral Efficiency at different SNR

Table 1. Comparative study of spectral Efficiency

Network	Spectral Efficiency (bits per sample)
Standard ad-hoc	0.82
2x2 MIMO ad-hoc	1.56
4x4 MIMO ad-hoc	3.82

4 Conclusions

We have simulated various MIMO ad-hoc networks with different combination of transmitting and receiving antennas. From the simulated result it is found that the spectral efficiency of the MIMO channel with more number of transmitting and receiving antennas increases by a factor of number of antennas in the transmitter and receiver. The spectral efficiency of the channel increases with the number of antennas and this become constant for a certain number of antennas. As the nodes of the mobile ad-hoc networks are battery operated so there is a limitation of the SNR as well as number of antennas of the system. It is found that the maximum 15 number of antenna is sufficient to get maximum spectral efficiency at 7db SNR.

References

1. Iwata, et al.: Scalable Routing Strategies for Ad hoc Wireless Networks. *IEEE Journal on Selected Areas in Communication, Special Issues on Ad-hoc Networks*, 1369–1379 (August 1999)
2. Das, S.R., Castaeda, R., Yan, J., Sengupta, R.: Comparative Performance Evaluation of Routing Protocol for Mobile Ad hoc Networks. In: *Proceedings of 7th International Conference on Computer Communication and Networks (IC3N)*, pp. 153–161 (October 1998)
3. Tsigros, A., Hass, Z.J.: Multi path routing in Presence of Frequent Topological Changes. *IEEE Communication Magazine*, 132–138 (November 2001)
4. Telatar, I.E.: Capacity of multi-antenna Gaussian Channels. *European Trans. Commun.* 10(3), 586–595 (1999)
5. Shin, H., Lee, J.H.: On the capacity of MIMO wireless channels. *IEICE Trans. Commun.* E87-B(3), 671–677 (2004)
6. RFC 802.11n Standard for MIMO communication. www.802.11n.org standard
7. Wang, Z., Giannakis, G.B.: Outage mutual information of space-time MIMO channels. *IEEE Trans, Inform. Theory* 50(4), 657–662 (2004)
8. Ye, S., Blum, R.: Optimized signaling for MIMO interferences systems with feedback. *IEEE Trans. On signal processing* 51(11), 2839–2848 (2003)
9. Lozano, A., Tulino, A., Verdu, S.: High-SNR power offset multi antenna communication. *IEEE Trans, Inform. Theory* 51(12), 4134–4151 (2005)
10. Hsu, R.C.J., Tarighat, A., Shah, A., Sayed, A.H., Jalali, B.: Capacity Enhancement in coherent optical MIMO (COMIMO) Multimode Fiber Links. *IEEE Communications letters* 10(3) (March 2006)
11. Elouadi, E.H., Jraifi, A., Saidi, E.H.: Power optimization in MIMO ad-hoc network. *IEEE Communications letters* 978-1-4244-3757-3/09

Optimization of Multimedia Packet Scheduling in Ad Hoc Networks Using Multi-Objective Genetic Algorithm

R. Muthu Selvi and R. Rajaram

Department of Information Technology,
Thiagarajar College of Engineering, Madurai
Tamilnadu, India
{rmsit, rrajaram}@tce.edu

Abstract. This paper presents a new approach to schedule the variable length multimedia packets in Ad hoc networks using Multi-Objective Genetic Algorithm (MOGA). Earlier the algorithm, called Virtual Deadline Scheduling (VDS) attempts to guarantee m out of k job instances (consecutive packets in a real-time stream) by their deadlines are serviced. VDS is capable of generating a feasible window constrained schedule that utilizes 100% of resources. However, when VDS either services a packet or switches to a new request period, it must update the corresponding virtual deadline. Updating the service constraints is a bottleneck for the algorithm which increases the time complexity. MOGA overcomes the problem of updating the service constraints that leads to the increased time complexity. The packet length and the number of packets to be serviced are the two conflicting criteria which are affecting the throughput of scheduling. Using Multi Objective Genetic Algorithm (MOGA), a trade off can be achieved between the packet length and the number of packets to be serviced. MOGA produces an optimized schedule for the multimedia packets.

Keywords: Multi Objective genetic algorithm, window constrained scheduling, multimedia streaming, variable packet length.

1 Introduction

Recently, we are able to see an increasing interest in the field of multimedia. Multimedia applications include Video-on-demand, video authoring tools, news broadcasting, video conferencing, digital libraries and interactive video games. In these applications, the presence of many consecutive non-ordered multimedia packets in video and audio streams sent over a network might result in significant delay while buffering and receiving it. Arranging the packets efficiently can be done by window constrained scheduling. Window-constrained scheduling is a form of weakly-hard [1, 2] service, in which a minimum number of consecutive job instances (e.g., periodic tasks or consecutive packets in a real-time stream) must be processed by their deadlines in every finite window. Scheduling a given number of variable length packets effectively in order to achieve a maximal network throughput i.e. Optimal Packet Scheduling is well known as a NP-hard problem [3].

West et al have developed Dynamic Window Constrained Algorithm (DWCS) which attempts to guarantee no more than k out of a fixed window of m deadlines are missed for consecutive job instances as long as the total utilization of all required job instances does not exceed 100%. However, DWCS is only capable of guaranteeing a feasible schedule when all jobs have the same request period. [4, 5, 6]

West et al have improved DWCS by combining both the request deadlines and window constraint of jobs together to determine the virtual deadline. Virtual deadlines are adjusted dynamically as the urgency of servicing a job changes. VDS attempts to service m out of k job instances by their virtual deadlines that may be some finite time after the corresponding real-time deadlines. Notwithstanding, VDS is capable of outperforming DWCS and similar algorithms, when servicing jobs with potentially different request periods. Additionally, VDS is able to limit the extent to which a fraction of all job instances are serviced late [7]. Unlike DWCS, VDS is capable of scheduling jobs with different request periods. Both, Low bit rate cellular systems and high bit rate local area networks are more prone to bit errors. The quality degradation is observable even at the transport and application layers in the form of increased packet loss. Specifically for the applications like multimedia streaming where timely delivery of data is required, both effects are harmful. Large packets are more likely than small packets to be discarded due to bit errors. On the other hand, small packets lead to higher proportional protocol header overhead. Therefore packet size optimization is an essential research problem in Ad hoc Networks. [9]

Jari Korhonen and Ye Wang have shown in their work that application level packet size optimization could facilitate efficient usage of wireless network resources, improving the service provided to all end users sharing the network. [10]

Most of the real world problems have multiple objectives which should be simultaneously optimized to get optimum result of the problem. In certain cases, these objectives can be optimized separately, but in most cases each objective has some epistasis, where perfect combination of these objectives is necessary for an optimum result. But, in many if not most real world problems, these relationships are contradictory to each other. If one of the objectives is optimized then things become worse in another, so we have to find a trade off between the objectives so that we find a solution that is acceptable to the user. The definition of acceptability depends upon the scenario for which the problem is being optimized. [11]

In this paper, we propose a novel Multi-objective Genetic Algorithm (MOGA) to approach the fast convergence. Due to the fast convergent property of MOGA, it is possible to schedule the variable length packets efficiently in high speed networks. In our work, we use MOGA for the optimal packet scheduling with the length of the packets and the number of packets to be serviced as schedule parameters which are the conflicting objectives. While scheduling the packets, it is required to decide which packets to give more priority i.e. (Co-scheduling) and how to reduce the delay among the co-scheduled packets. MOGA solves these two issues.

This paper is organized as follows: In Section 2 we define the problem. Section 3 briefs the methodology MOGA and Virtual deadline. Section 4 presents the implementation of MOGA and Section 5 presents our comparison results. Section 6 contains our conclusion and future work.

2 Problem Definition

Virtual Deadline Scheduling provides window-constrained service guarantees to jobs with potentially different request periods, while still maximizing resource utilization. VDS attempts to service m out of k job instances by their virtual deadlines that may be some finite time after the corresponding real-time deadlines. But still VDS faces many problems like infeasible window problem, window constraint violations etc.

To overcome the above problems of VDS we have followed Multi-objective Genetic Algorithm (MOGA) methodology to produce an optimized scheduling pattern. For multiple objective problems, the objectives are generally conflicting, preventing simultaneous optimization of each objective. Many, or even most, real engineering problems actually do have multiple objectives, i.e., minimize cost, maximize performance. In our project the conflicting criteria are the length of the packets and the minimum number of packets that must be scheduled for service by their deadlines. In our work, we use Multi-objective Genetic Algorithm to find a trade off between the length of the packets and a $(k_i - m_i)$ (i.e.) minimum number of packets that must be scheduled for service by their deadlines and to find out an optimized scheduling pattern for the multimedia packets. k_i denotes the maximum no. of packets to be serviced in current window and m_i denotes the minimum no. of packets to be serviced in current window.

3 Methodology

3.1 Multi-Objective Genetic Algorithm

3.1.1 Overview

Being a population-based approach, GA is well suited to solve multi-objective optimization problems. A generic single-objective GA can be modified to find a set of multiple non-dominated solutions in a single run. The ability of GA to simultaneously search different regions of a solution space makes it possible to find a diverse set of solutions for difficult problems with non-convex, discontinuous and multi-modal solutions spaces.

Multi-objective Genetic Algorithm (MOGA) optimization problems are very difficult because they often deal with conflicting objectives. Many different approaches have been applied to MOGA problems. Aggregation-based approaches use a weighted sum of the objective values as the new objective in a single-objective optimization problem. Criterion-based approaches consider only one objective of a MOGA problem at a time. In the simplest case, the objectives are ranked in order of importance, optimizing each one in turn without degrading the values of the previous objectives.

3.1.2 Mathematical Model

Let $x_0, x_1, x_2 \in F$, and F be a feasible region. x_0 is called the Pareto optimal solution in the minimization problem if the following conditions are satisfied. If $f(x_1)$ is

said to be partially greater than $f(x_2)$, i.e. $f_i(x_1) \geq f_i(x_2), \forall i=1, 2, \dots, n$ and $f_i(x_1) > f_i(x_2), \exists i=1, 2, \dots, n$, Then x_1 is said to be dominated by x_2 .

If there is no $x \in F$ such that x dominates x_0 , then x_0 is the Pareto optimal solutions.

3.1.3 Steps in MOGA

Step 1: Generate Initial Population

Initial solutions are randomly generated and these initial solutions form the first population.

Step 2: Record Pareto Optimal Solutions

Calculate the objective values of chromosomes in the population and record the Pareto optimal solutions.

Step 3: Calculate Objective Value

The total objective function is constituted of the linear combination of objective functions. And the weights are randomly assigned. For a solution x , the objective function in the study is represented as follows:

$$f(x) = w_1 \cdot f_1(x) + w_2 \cdot f_2(x)$$

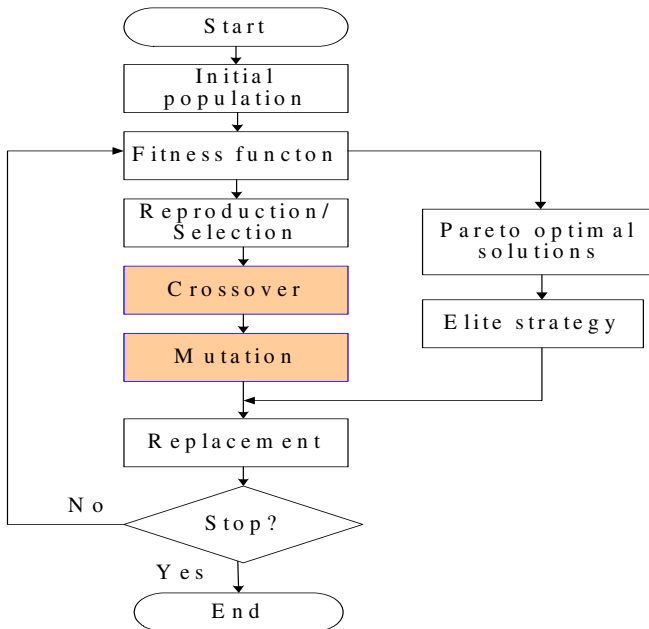


Fig. 1. Steps of MOGA

Step 4: Evaluate Fitness

The original concept of fitness is the larger the better because solutions with larger fitness tend to propagate to the next generation. In this paper, minimization of objectives is

considered. Hence it contradicts the original idea of fitness. A transformation should be made to reverse the minimization to maximization. For a solution x , its fitness equals to the maximal objective value in the generation minus itself. The formula is listed in the following.

$$\text{fitness}(x) = f_{\max}(x) - f(x)$$

Step 5: Reproduction / Selection

New offspring for next generation of MOGA are formed by selection method according to their fitness values. The selection process is known as evolution operator, directing a MOGA search toward promising region in the search space. There are various types of selection strategies such as elitist selection strategy, stochastic tournament selection strategy, and so on, in order to improve population to more evolved population (with better fitness).

These selection strategies provide the driving force in a MOGA and the selection pressure is critical in it. At one extreme, the search will terminate prematurely; while at the other extreme progress will be slower than necessary. It seems that there are two important issues in the evolution process of genetic search population diversity and selective pressure. These factors are strongly related: an increase in the selective pressure decreases the diversity of the population, and vice versa. In other words, strong selective pressure supports the premature convergence of the MOGA search; a weak selective pressure is recommended at the end in order to exploit the most promising regions of the search space. Thus it is important to strike a balance between these two factors.

Step 6: Crossover

Crossover [12] may be a main genetic operator. Crossover operator selects two chromosomes at a time and generates offspring by combining both individual's features. A simple way to achieve crossover is to choose a random cut-point and generate the offspring by combining the segment of one parent to the left of the cut-point with the segment of the other parent to the right of the cut-point.

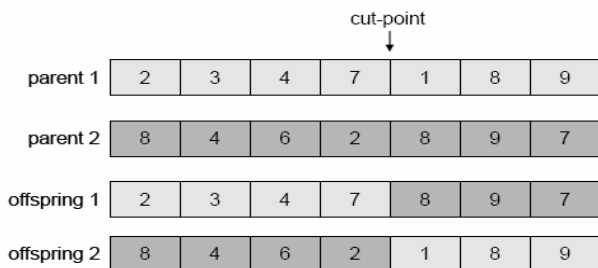


Fig. 2. Crossover

Step 7: Mutation

Mutation [12] is usually used as a background operator, which produces spontaneous random changes in various individuals. A simple way to achieve mutation would be to alter one or more genes.

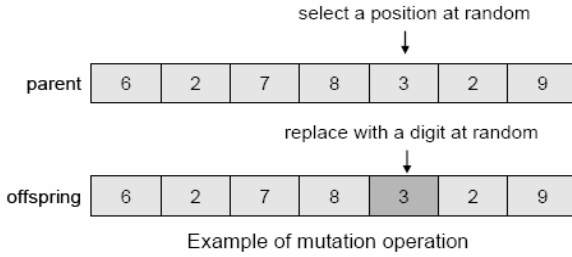


Fig. 3. Mutation

Step 8: Elite Strategy

The elite strategy retains the top k solutions in order to keep quality solutions in each generation.

Step 9: Replacement

The new population generated by the previous steps. Update the old population.

Step 10: Update Pareto Optimal Solutions.

Search the Pareto optimal solutions in the new population and update the old Pareto optimal solutions with new ones.

Step 11: Stopping rule

If the number of generations equals to the pre-specified number then stop, otherwise go to step 3.

3.2 Virtual Deadline

In the context of Packet Scheduling, Virtual Deadline refers to some finite time after the corresponding real-time deadlines. A packet’s virtual deadline with respect to real-time, t , is shown in Equation (1) [7], T'_i is the remaining time in the current request period for packet P_i . $(t + T'_i - T)$ is the start time of current request period, which we denote as $t_r(t)$ for brevity. Similarly, (m'_i, k'_i) represent the current window-constraint at time, t . This implies that window-constraints change dynamically, depending on whether or not a packet instance is serviced by its deadline.

At this point, it is worth outlining the intuition behind a packet’s virtual deadline. If at time t , P_i ’s current window-constraint is (m'_i, k'_i) , then $m_i - m'_i$ out of $k_i - k'_i$ job instances have been serviced. There are still m'_i packet instances that need to be serviced in the remaining time in the current window, which is $k'_i T_i$. If

one instance of P_i is serviced every interval $\frac{k'_i T_i}{m'_i}$, then m'_i packets will be serviced in the current remaining window-time, $k'_i T_i$.

$$Vd_i(t) = \frac{k_i T_i}{m_i} + t_r(t) \tag{1}$$

where $t_r(t) = t + T_i' - T_i$

Fig 4 gives an example of the virtual deadline calculation. We can see that, if a packet's current window-constraint does not change within a request period, its virtual deadline will not change either. In this example more than one job instance can be served in one request period which is called as the relaxed window-constrained model.

Let C_i (service time) =1, T_i (request period) =4, $m_i=2$, $k_i=3$, m out of k packets to be serviced.

Current time, $t=16$, $Vd(16) = 20$

- $Vd(t=16) = (k' * T / m') + (t + T' - T) = (2*4/2) + (16+4-4) = 20$
- $Vd (t=17) = (2*4/2) + (17+3-4) = 20$
- $Vd (t=18) = (2*4/1) + (18+2-4) = 24$
- $Vd (t=19) = (2*4/1) + (19+1-4) = 24$

•Virtual deadline, $Vd(t)$, remains at 24 until start of next window, at $t=24$, because $m'=0$ at $t=20$.

4 Implementation

We developed a Linux kernel module which schedules the real time packets using MOGA. We used the Linux kernel 2.6.18.

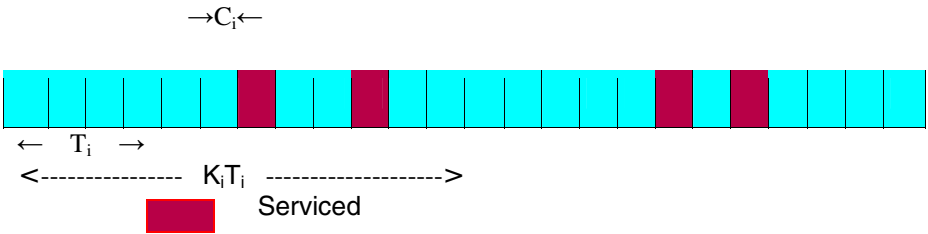


Fig. 4. Virtual Deadline Calculation

4.1 Wireshark

We collected the packets using Wireshark 1.2.1. Wireshark is a network packet analyzer which analyses the real-time network packets and gives information about the packets such as protocol, header information, packet length, arrival time of the packet etc. We used the packet length and arrival time of the packet from Wireshark.

4.2 MOGA

We used real coding for encoding the packets. We used a population size of 20. The roulette wheel selection is applied in MOGA. 200 generations are required for reaching

the optimal value. Two point cross over and shift mutation operators are used in the experiment. The cross over rate and the mutation rate are 0.8 and 0.01 respectively. We used weighted sum to combine multiple objectives into single objective. For example:

$$f(x) = w_1 f_1(x) + w_2 f_2(x) + \dots + w_n f_n(x)$$

where $f_1(x), f_2(x), \dots, f_n(x)$ are the objective functions and w_1, w_2, \dots, w_n are the weights of corresponding objectives that satisfy the following conditions. $w_i \geq 0 \quad \forall i=1,2,\dots,n$

$$w_1 + w_2 + \dots + w_n = 1$$

The fitness function of MOGA for our work is given by

$$f(L, ki - mi) = \max \{ k1 t_a + k2L + k3(ki - mi) \}$$

where L denotes the packet length, t_a represents the arrival time of the packet and $K_i - m_i$ gives the maximum number of packets to be serviced and k1, k2 and k3 are coefficients for weighted sum

- k1 is a positive constant
- k2 = 66.67% of k1
- k3 = 7% of k1

5 Experimental Results

5.1 Violation of Window Constraint

We collected the packets from a multimedia presentation through Wireshark and scheduled the variable length packets using VDS and MOGA. We observed the following:

For some packet instances, there is a violation of window constraints (i.e.) (Servicing the various packet instances within the deadline) in the case of VDS whereas the schedule produced by MOGA, there is no such violation of window constraint.

According to the window constraint for P2 in Table 1, out of 9 request periods of P2 at least 5 request periods have to be satisfied. Fig 5 shows that VDS could not handle the processes with varying service times and request period, Only 4 request periods are serviced and the window constraint of J2 is violated. Fig 6 shows the result of scheduling using MOGA. Here the window constraints of both the packets are satisfied.

Table 1. Window constraints for packets

Packet	C	T	m	K
P1	2	3	2	3
P2	1	1	5	9

P2	P2	P1	P2	P2	P1	P1	P1	P1
----	----	----	----	----	----	----	----	----

Fig. 5. Schedule produced by VDS. P1 (shaded cell) shows where P2 violates the window constraint.

P2	P1	P1	P1	P1	P2	P2	P2	P2
----	----	----	----	----	----	----	----	----

Fig. 6. Schedule produced by MOGA

5.2 Average Waiting Time of Packets

We have tabulated the average waiting time of packets for VDS and MOGA in Table 2. The average waiting time of packets scheduled using MOGA is very less than that of VDS. Since the throughput depends on the average waiting time, the performance is improved when we use MOGA for scheduling the packets. The comparison is shown in Fig 7.

5.3 Updating Service Constraints

In VDS, the virtual deadlines are updated using the equation (1). West et al have assumed that scheduling decisions and service constraint adjustments are made once every timeslot, Δ and Δ represents a unit time-slot. The algorithmic complexity of the VDS algorithm is a linear function of the number of jobs needing service, in the worst case [7]. As number of packets increases, updating the service constraints is a major bottleneck. MOGA does not require any service constraint adjustments.

Table 2. Comparison of average waiting times of packets for VDS and MOGA

Number of Packets	Average waiting time of packets (using VDS) (ms)	Average waiting time of packets (using MOGA) (ms)
5	3.6	1.6
10	4.2	2.2
15	4.9	1.9
20	5.1	2.3

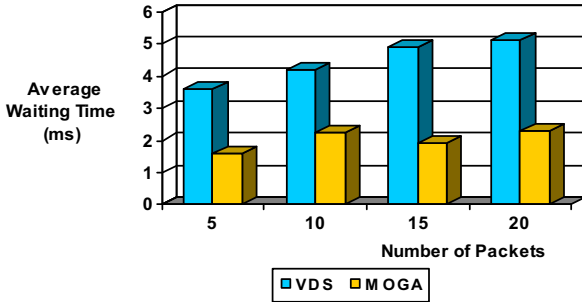


Fig. 7. Comparison of average waiting time of packets for VDS and MOGA

6 Conclusion and Future Work

Though the objectives variable packet length and the minimum number of packets that must be serviced for a given window are the conflicting objectives, MOGA is able to find an optimal solution and it produces an optimized scheduling pattern. Updating the service constraints repeatedly is a major bottleneck in VDS. It also increases the time complexity. But this difficulty is overcome by MOGA. Hence, MOGA is suitable for scheduling multimedia packets.

In future, we will include more parameters like loss tolerance which affects the throughput. We will hybridize the algorithm with local search technique, Simulated Annealing.

References

1. Bernat, G., Burns, A., Llamosi, A.: Weakly-hard real-time systems. *IEEE Transactions on Computers* 50(4), 308–321 (2001)
2. Bernat, G., Cayssials, R.: Guaranteed on-line weakly-hard real-time systems. In: *Proceedings of the 22nd IEEE Real-Time Systems Symposium* (2001)
3. Waa, R.-G., Lee, Y., Shu, S.T., Tseng, H.-W., Chuang, M.-H., Waung, Y.H.: Application of hardware Architecture of Genetic Algorithm for optimal packet scheduling. *International Journal of Fuzzy Systems* 10(3), 202–206 (2008)
4. West, R., Poellabauer, C.: Analysis of a window constrained scheduler for real-time and best-effort packet streams. In: *Proceedings of the 21st IEEE Real-Time Systems Symposium* (2000)
5. West, R., Schwan, K., Poellabauer, C.: Scalable scheduling support for loss and delay constrained media streams. In: *Proceedings of the 5th IEEE Real-Time Technology and Applications Symposium* (1999)

6. West, R., Zhang, Y., Schwan, K., Poellabauer, C.: Dynamic window-constrained scheduling of real-time streams in media servers. *IEEE Transactions on Computers* 53(6), 744–758 (2004)
7. Zhang, Y., West, R., Qi, X.: A virtual deadline scheduler for window constrained service guarantees, Tech. Rep. 2004-013, Boston University (2004)
8. Zitzler, E.: Thiele: Multi Objective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Trans. on Evolutionary Computation* 3(4), 257–271 (1999)
9. Al-Saber, N.A., Oberoi, S., Rojas-Cessa, R., Zivavras, S.G.: Concatenating Packets in Variable-Length Input-Queued Packet Switches with Cell-Based and Packet-Based Scheduling. In: *Proceedings of the 32nd International Conference on Sarnoff symposium* (2009)
10. Korhonen, J., Wang, Y.: Effect of Packet Size on Loss Rate and Delay in Wireless Links. In: *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC 2005)*, pp. 1608–1613 (2005)
11. Savic, D.: Single-objective vs. Multi-objective Optimization for Integrated Decision Support. In: *Proceedings of the First Biennial Meeting of the International Environmental Modeling and Software Society* (2002)
12. Goldberg, D.E.: *Genetic Algorithms in Search, Optimization*. In: *Machine Learning*, 1st edn., Addison-Wesley Professional, Reading (1989)
13. Deb, K.: *Multi-objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Chichester (2001)
14. <http://lxr.linux.no>

TRIDNT: Isolating Dropper Nodes with Some Degree of Selfishness in MANET

Ahmed M. Abd El-Haleem¹, Ihab A. Ali², Ibrahim I. Ibrahim³,
and Abdel Rahman H. El-Sawy⁴

¹ Assistant Lecture, Communication Department, Faculty of Engineering,
Helwan University, Helwan
ahmed_abdelkhalig@h-eng.helwan.edu.eg

² Associate Professor, Communication Department, Faculty of Engineering,
Helwan University, Helwan
ehab_ali02@h-eng.helwan.edu.eg

³ Professor, Communication Department, Faculty of Engineering, Helwan University, Helwan
ibrahim_ibrahim02@h-eng.helwan.edu.eg

⁴ Professor, Communication Department, Faculty of Engineering, Helwan University, Helwan
abdelrahman_alsawy@h-eng.helwan.edu.eg

Abstract. In Mobile ad-hoc network, nodes must cooperate to achieve the routing purposes. Therefore, some network nodes may decide against cooperating with others; selfish nodes; to save their resources. Also these networks are extremely under threat to insider; malicious nodes; especially through packet dropping attacks.

In this paper, we design a novel monitoring and searching scheme to detect and isolate the dropper nodes in ad-hoc networks, called TRIDNT (Two node-disjoint Routes scheme for Isolating Dropper Node in MANET). TRIDNT allows some degree of selfishness to give an incentive to the selfish nodes to declare itself to its neighbors, which reduce the misbehaving nodes searching time. In TRIDNT two node-disjoint routes between the source and destination are selected based on their trust values. We use both DLL-ACK and end-to-end TCP-ACK to monitor the behavior of routing path nodes: if a malicious behavior is detected then the path searching tool starts to identify the malicious nodes and isolate them. Finally our scheme reduces the searching time of malicious nodes, and avoids the isolated misbehaving node from sharing in all future routes, which improve the overall network throughput.

Keywords: Ad Hoc Network, Secure Routing Protocol, network security.

1 Introduction

Mobile Ad Hoc Network (MANET) is an infrastructure-less network, consisting of a set of mobile nodes without any support of base stations or access points. The mobile nodes are free to change their position with any speed and at any time, and they play the role of terminals and routers allowing hop by hop communication among nodes outside wireless transmission range. For lack of network infrastructure, the nodes

have to communicate cooperatively. Cooperation at the network layer means routing and forwarding packets. Some nodes may deviate from the protocol for selfish or malicious reasons, these nodes are called misbehaving nodes. Selfish nodes wish to use system services while taking an advantage of saving their resources by deviating from regular routing and forwarding. Malicious nodes wish to mount an attack to either a specific node or the network as whole. Both selfish and malicious nodes disrupt the routing protocol operation and reduce the network throughput. This brings up the need for secure routing protocols, where the Routing protocols must cope with such selfish and malicious behavior.

Several routing protocols have been proposed in the literature (see, e.g., [1], [2], [3]). These focus mainly on efficiency issues such as scalability with respect to network size, traffic load, mobility, and on the adaptability to network conditions such as link quality and power requirements. Some of the proposed routing algorithms also address security issues by using cryptographic tools to secure the routing protocol messages (e.g., [4], [5], [6], [7] for a survey, see [8], [9]). Recently, a new class of routing protocol has been proposed, namely trust based routing as in [10]. Trust based routing protocols consist of two parts: a routing part and a trust model, for a survey see [11]. Routing decisions are made according to the trust model. The trust routing protocols have to be able to identify trustworthy nodes and find a reliable and trustworthy route from sender to destination node. This has to be realized within a few seconds or better tenths of seconds, depending on the mobility of the nodes and the number of hops in the route. Most of the existing trust based routing protocols uses continuous promiscuous monitoring of the neighbors; which violate the TCP protocol rules.

This paper focuses on Packet Dropping Attack, and presents a novel routing algorithm resistant to various packet dropping scenarios. Here, the malicious node tends to threaten network throughput through the use of packet dropping attack. This kind of attack could be even worse when supported by the malicious node sending link-layer acknowledgements to neighbor nodes to delay the detection of the attack and hence further decrease the throughput. In this paper, four packet dropping scenarios are considered. In Inclusive Packet Dropping, the malicious node simply drops all received data link layer (DLL) PDU's while positively acknowledging them. This attack is also called Black Hole attack [12], [13], [5]. Periodic Packet Dropping is used by malicious nodes to drop a small fraction of incoming DLL PDU's once per retransmission time out, a variant of JellyFish (JF) attack reported in [12], [13], [14]. In Frequent Packet Dropping, the malicious node may possibly drop a fraction of incoming DLL PDU's on a random basis. In Selective Packet Dropping, the malicious node drops only these PDU's coming from specific source(s), going to specific destination(s), or following a specific route. The last two attacks are called Gray Hole attack. In all packets dropping attacks scenarios, the overall network throughput will deteriorate [12].

In our proposed scheme we establish two node-disjoint routes between the source and destination nodes, these routes have the highest path trust values; to route around misbehaving nodes; one is marked as primary and the other as secondary. Unlike all previous research efforts made to tolerate Packet Dropping Attacks, our work allow some degree of node selfishness; to save their resources partially; and detect the malicious activity faster. We use both DLL-ACK and end-to-end TCP-ACK as monitoring

tools; without continuous promiscuous monitoring of the neighbors; and when detecting a malicious activity a new path searching technique is used to identify the malicious or compromised nodes in the routing path and isolate them. Based on this claim, the proposed scheme avoids the isolated misbehaving node from sharing in all future routes, resulting in an improved overall throughput performance for the network.

The rest of the paper is organized as follows. Section 2, describes the related work. The network assumptions and the TRIDNT operation are presented in Section 3. Finally we conclude our work and discuss our plan for future work in section 4.

2 Related Work

In [15] Marti et al. proposed a mechanism called as watchdog and pathrater on DSR to detect the misbehaving nodes in MANETs. The approach introduces two extensions to DSR: A watchdog detects misbehaving nodes, by maintaining a buffer of transmitted packets and overhearing of other node forwarding. It compares each overheard packet with the packets in the buffer to see if there is a match. If so, the packet in the buffer is removed and forgotten by the watchdog, since it has been forwarded on. If a packet has remained in the buffer for longer than a certain timeout, the watchdog increments a failure tally for the node responsible for forwarding on the packet. If the tally exceeds a certain threshold bandwidth, it determines that the node is misbehaving and sends a message to the source notifying it of the misbehaving node. A pathrater avoids routing packets through the detected malicious nodes. Each node estimates a link metric with respect to the reliability of links and knowledge about misbehaving nodes. A node assigns this metric to every other known node and periodically updates the metric. The downside of their method is that they cannot distinguish the misbehaving nodes from node failures. An honest node can easily be rated malicious if the transmission breaks up.

CONFIDANT [16] is a protocol which also attempts to detect the malicious nodes in ad hoc networks. Monitor, Reputation System, Path Manager and Trust Manager are the main components of CONFIDANT protocol. For each packet a node forwards, the monitor on that node attempts to ensure that the next-hop node also forwarded the packet correctly (overhearing). When the monitor detects an anomaly, it triggers action by the reputation system, which maintains a local ratings list. These lists are potentially exchanged with other nodes; the trust manager handles input from other nodes. Finally, the path manager chooses paths from the node's route cache based on a blacklist and the local ratings list. CONFIDANT has scalability problems with the number of nodes. The tables maintained by the reputation system of each node may become huge. Also, in scenarios with very high mobility, the overhead can increase considerably.

In [17] Balakrishnan et al, propose a scheme of TWOACK to prevent selfishness in mobile ad hoc networks. They proposed two network-layer acknowledgment-based schemes, termed the TWOACK and the S-TWOACK schemes, which can be simply added-on to any source routing protocol. When a node forwards a packet, the node's routing agent verifies that the packet is received successfully by the node that is two hops away on the source route. This is done through the use of a special type of acknowledgment packets, termed TWOACK packets. TWOACK packets have a very

similar functionality as the ACK packets. A node acknowledges the receipt of a data packet by sending back a two-hop TWOACK packet along the active source route. If the sender/forwarder of a data packet does not receive a TWOACK packet corresponding to a particular data packet that was sent out, the next-hop's forwarding link is claimed to be misbehaving and the forwarding route broken. Based on this claim, the routing protocol avoids the accused link in all future routes, resulting in an improved overall throughput performance for the network. The S-TWOACK (Selective-TWOACK) scheme is a derivative of the basic TWOACK scheme, aimed at reducing the routing overhead caused by excessive number of TWOACK packets. The basic drawback of this scheme is that it can't determine exactly which node is the misbehaving node; it only marks the link interconnecting the two nodes as misbehaving link and tries to avoid using this link in future.

Muhammad Zeshan et al, [18] proposed a two folded approach, to detect and then to isolate a malicious node causing packet dropping attacks. First approach will detect the misbehavior of nodes and will identify the malicious activity in network. When a Source node forwards any packet to the Destination through a route, all intermediate nodes will send back an ACK packet to its source node. If the Source node doesn't receive the ACK from any intermediate node, it will send again its packet for Destination after a specific time but if again this activity was observed, Source node will broadcast a packet to declare the malicious activity in the network. Then upon identification of misbehaving nodes in network other approach will isolate the malicious node from network. All nodes which lie in the transmission range of active route and also the nodes which are on the active route become in promiscuous listening mode and count number of packet coming into and going out of the nodes of active route. Each node in this range maintains a list of sent and dropped packets and when number of dropped packets by a particular node exceeds a certain threshold, the monitoring node in that range declares that node as misbehaving node. The basic drawback of this scheme is, nodes cooperate together to obtain an objective opinion about another node's trustworthiness, which give the misbehaving node the chance to falsely report the value of trust score (False Misbehaviour).

3 The Proposed Algorithm

In this section we describe our solution to address the Packet Dropping Attack in MANETs. Our proposed scheme makes the first effort to distinguish between the malicious and selfish node, and allow a controlled degree of node selfishness. The proposed monitoring tool detects the malicious activity and then the path searching tool identifies the malicious or compromised nodes in the network and isolates them, and the proposed routing algorithm routes around the misbehaving node.

3.1 Network Model and Assumption

In this work, we assume that the MANET nodes are situated in a bounded two-dimensional space, within which they are free to move, and a bi-directional communication symmetry on every link between the nodes. For simplicity we also assume that the destination-node is non-malicious, and any routing path contains at most one malicious node.

Mobile nodes in MANETs often communicate with one another through an error-prone, bandwidth-limited, and insecure wireless channel. We are not concerned with the security problem introduced by the instability of physical layer or link layer. We only assume that: (1) Each node in the network has the ability to discover all of its neighbors; (2) Each node in the network can broadcast some essential messages to its neighbors with high reliability; (3) Each node in the network uses its MAC address as a unique identifier (node ID); (4) Each node in the network have a black list containing the misbehaving nodes, a trust table containing the learned network nodes' trust value; and a Data Packet Information (DPI) cache to store information about the received and processed data or TCP-ACK packets.

3.2 Operation of TRIDNT

The misbehaving node may be a selfish or malicious node, our approach allows some degree of selfishness for nodes to save their resources (e.g. battery power; where nodes behave differently based on their energy levels. When the energy lies between full energy E and a threshold E_s , the node behaves properly. For an energy level lower than the threshold E_s , it uses its energy for transmissions of its own packets) to reduce our search to find malicious node only. A new field is inserted in the Hello packet containing the selfishness status. Each node use this filed in Hello packet to inform its direct neighbor nodes about its selfishness status, if it is in selfish mode all neighbor nodes will:

1. Don't forward any Route Request (RREQ) packet to it.
2. Remove it from the active routes, which it is an intermediate node on it, and send Route Error (RERR) packet to the sources to establish new routes.
3. Forward to/from it the packets which contain it as destination/source address.

The selfish node neighbors will restrict its selfishness behavior by a time (selfishness interval) threshold, and a repetition (number of selfishness times) threshold.

In our approach we use the DLL-ACK and the end to end TCP-ACK as a monitoring tool to monitor the behavior of a path, then use a path searching tool to search the misbehaving path to find the misbehaving node, and then put the misbehaving node ID in the black list to isolate it.

3.2.1 Route Discovery Phase

In route discovery phase the source node S flooding a route request (RREQ) for the destination node D in the network to establish two node disjoints paths as in AOMDV [19], or multipath DSR [20], and then waits for a route reply (RREP). The RREQ packet contains a list of unwanted nodes, which the source node doesn't want them to be a members on the discovered route temporarily.

When an intermediate node receives the first copy of a RREQ packet, it sets up a reverse path to the source using the previous hop of the RREQ as the next hop on the reverse path, and updates the route trust value in the RREQ packet. In addition, if there is a valid route available for the destination, it unicasts a RREP back to the source via the reverse path; otherwise, it re-broadcasts the RREQ packet to the nodes

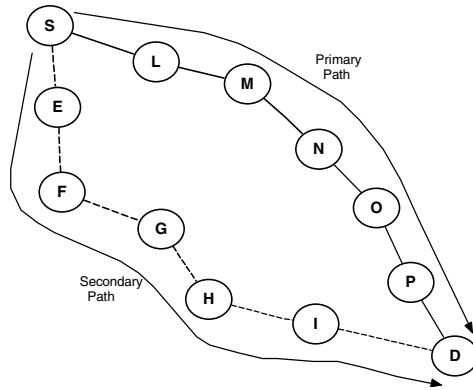


Fig. 1. Two node disjoint paths between S and D

not contained in the unwanted nodes list. Duplicate copies of the RREQ are immediately discarded upon reception at every node. When the destination node receive RREQ packet from multiple nodes, it select a two-node disjoint paths with the highest path trust value.

Then the destination node forms the two forward paths by unicasts two RREPs (contain the path trust value) back to the source along the selected two routing paths. As the RREP proceeds towards the source, it establishes a forward path to the destination at each hop. The source node mark the highest trusted route as primary used for data forwarding and the other as secondary used as a backup path, as shown in figure 1.

3.2.2 Data Forwarding and Monitoring Phase

In the Data forwarding phase the source node send its data packet over the primary path only and each node in the path store the received data packet information in its Data Packet Information (DPI) cache, then forward it to its downstream neighbor, and wait for a data link layer acknowledgment (DLL-ACK) from the neighbor node, if it did not receive data link layer acknowledgement; it concludes that this neighbor node should be down. In such case, the neighbor is excluded from the node's routing table until it becomes up. However the neighbor's trust rating doesn't change.

On the other hand the source node waits to receive the end to end TCP-ACK from the destination node via the primary and secondary paths; source node in monitoring mode:

Case I: if there is no malicious node in the primary and secondary paths, then the source node receive the TCP-ACK over the two routes (primary and secondary), as shown in figure 2. The source node sends a biggy back Positive Trust Update Message (PTUM) upon transmitting the next packet. If the node which sent this message received an acknowledgment from a neighbor node in the data link layer and through this neighbor in the transport layer, and receive PTUM message from the source node. Then all nodes in the primary path will update the trust value of each other and remove the information about the confirmed data packet from its DPI cache. Also

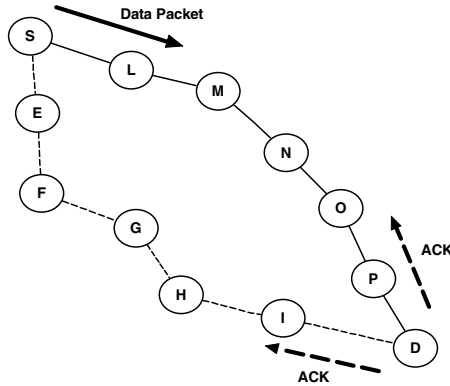


Fig. 2. The source node sends the data on the primary path only, and the destination replay with ACK on both primary and secondary paths

destination node will send a biggy back PTUM message when transmitting the next TCP-ACK packet to the source node, to update the trust value of nodes in the secondary path.

Case II: if the source node received an acknowledgment from a neighbor in the data link layer and receive an acknowledgment in the transport layer over the primary path only, even after retransmitting this message (TCP rules); it concludes that the neighbor node or one of its following nodes in the primary routing path may be malicious node trying to make blocking attack and send a faked TCP-ACK or there is a malicious node in the secondary routing path drop the TCP-ACK packet.

Case III: if the source node received an acknowledgment from a neighbor in the data link layer and did not receive an acknowledgment in the transport layer over the primary and secondary route paths, even after retransmitting this message (TCP rules). Then the source node knows that the data packet doesn't reach its destination, i.e. there is a malicious node in the primary path trying to make blocking attack.

In last two cases II and III the source node starts the malicious search phase, to find the malicious node.

3.2.3 Malicious Search Phase

In the malicious search phase the source node start a search to find the malicious node in the primary or secondary paths. In this phase the source node send a Malicious Search Packet (MSP); which contains information about the lost data packet; via the primary route towered the destination node.

The MSP packet is a high priority packet, and every node receive this packet compare its information with the data packet information's stored in its DPI cache, if it found a match (the node received this data packet and forward it to the next node) it will forward the MSP packet to the next node with overhearing to assure that the neighbor node will forward it. The node which found a mismatch will stop forwarding of MSP packet and generate a Malicious Detection Packet "MDP (detecting node ID,

detected node ID)" contain its ID and the malicious node ID; it is a high priority packet forwarded with overhearing. Also the node which found that its downstream node doesn't forward the MSP packet generates the MDP packet. The node generating the MDP packet forwards it in the opposite direction to the detected malicious node, toward the source or destination node.

Case I: if the primary path contains a malicious node, let node N be the malicious node. The source node sends the MSP packet to node L and overhears to be sure that node L will forward that packet. After comparison, node L forwards the MSP packet to M and overhears, then node M compares and forwards it to node N and overhears. The malicious node N has two choices:

- (1) It either, stops forwarding the MSP packet and report the destination node using MDP packet (N, M), that node M is the malicious node; node N deny the receiving of this data packet from node M. At the same time node M sure that it forward this data packet to node N and receive a DLL-ACK from node N, where it didn't overhear node N forward MSP packet, then node M report the source node that node N is the malicious node; using MDP packet (M, N) as shown in figure 3-a.
- (2) Or, it will forward the MSP packet to node O, then node O didn't find a match in its DPI cache, so it will send the MDP packet (O, N) to the destination node. At the same time the malicious node N can inform the source node that node O don't forward the MSP packet; send a MDP packet (N, O) as shown in figure 3-b.

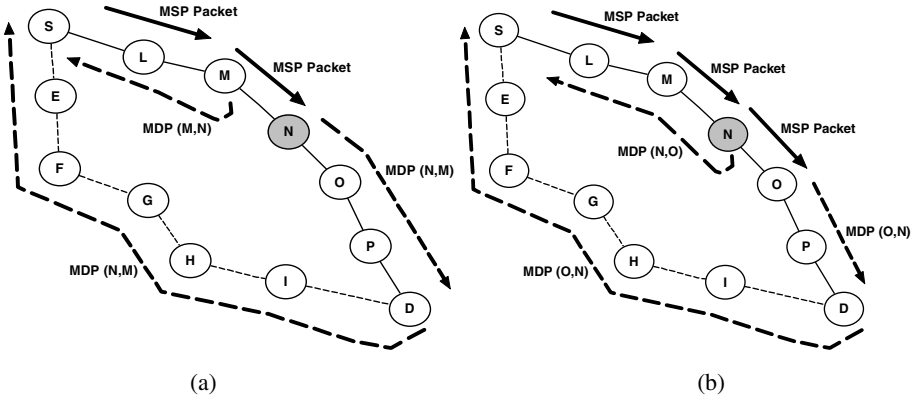


Fig. 3. Malicious node N (a) don't forward the MSP packet, (b) forward the MSP packet

In both cases when the source and destination nodes; and all nodes in the routing path; receive the MDP packet, they will update the trust value of both the detecting and detected node negatively and the trust value of other nodes in the routing path positively. Also the detecting node (node M or node O), will insert node N ID in its black list; regardless of its trust value; to reject any future cooperation between them (i.e. isolating the malicious node).

When the destination node receives the MDP packet it forwards it to the source via the secondary route. Finally the source node mark the secondary route as primary and start a route discovery phase to find a secondary node disjoint route not containing both the detecting and detected node on it.

Case II: if the secondary path contains a malicious node, let node H be the malicious node. The source node sends the MSP packet to node L, and node L forward it to node $M \rightarrow N \rightarrow O \rightarrow P \rightarrow$ to the destination.

When the destination node receives the MSP packet, it will be sure that there is no malicious node in the primary route, and then the destination node will modify the MSP packet to contain the information of TCP-ACK packet and forward it to node I via the secondary path. Node I forward it after comparison to node H (the malicious node) the malicious node H has the same two choices as in case I as shown in figure 4. When any node receives the MDP packet, it will update the trust values of both the detecting and detected node negatively, and the trust value of other nodes in the routing path positively. Also the detecting node (node I or node G), will insert node N ID in its black list; regardless of its trust value; to reject any future cooperation between them. When the destination node receives the MDP packet it forwards it to the source via the primary route. Finally the source node starts a route discovery phase to find a secondary node disjoint route not containing both the detecting and detected node on it.

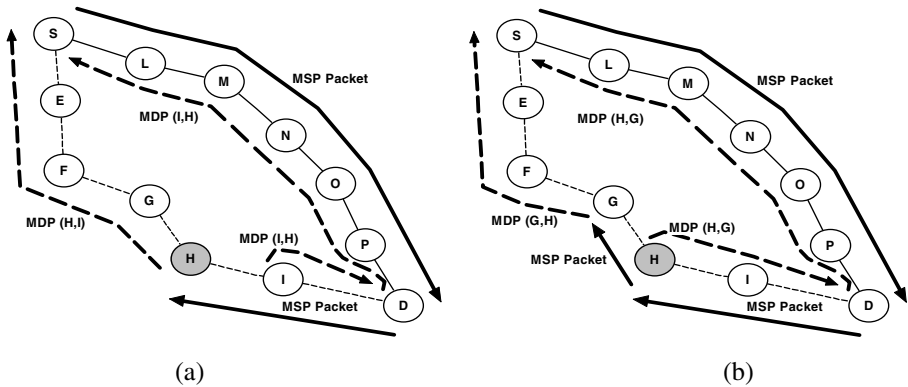


Fig. 4. Malicious node H (a) don't forward the MSP packet, (b) forward the MSP packet

Case III: if both the primary and secondary routes contain malicious nodes. The source node sends the MSP packet via the primary path and waits the MDP packet, if:

- (1) The primary path detecting node send the MDP packet (N, O) to the source node, on the other hand when the destination node receive an MDP packet (O, N) it will forward it to the source node via the secondary path. If the secondary path malicious node forward the MDP packet (O, N), then the source node receive the MDP packet (O, N) and mark the secondary path as primary and search for new secondary path don't contain both the detecting and detected nodes (M, N), as

shown in figure 5-a. Because the new primary route contain also a malicious node, then the source node don't receive TCP-ACK packet from the destination, so it start a new malicious search procedure to find the malicious node. When finding the new malicious node the source node marks the new secondary route as primary and search a new secondary route, and so on.

- (2) The primary path detecting node send the MDP packet (O, N) to the destination node, then the destination node send the MDP packet (O, N) to the source node via the secondary path. Figure 5-b shown that there is a malicious node in the secondary path drop the MDP packet (O, N), and don't reach the source node. When node I found that node H drops the MDP packet (O, N), it will send a new MDP packet (I, H) to the destination node. Then the destination node start searching for a new paths don't contains both detecting and detected nodes in the primary and secondary paths (O, N, I, H). When the source node receives the RREQ from the destination node; it concludes that there are malicious nodes in both old primary and secondary paths.

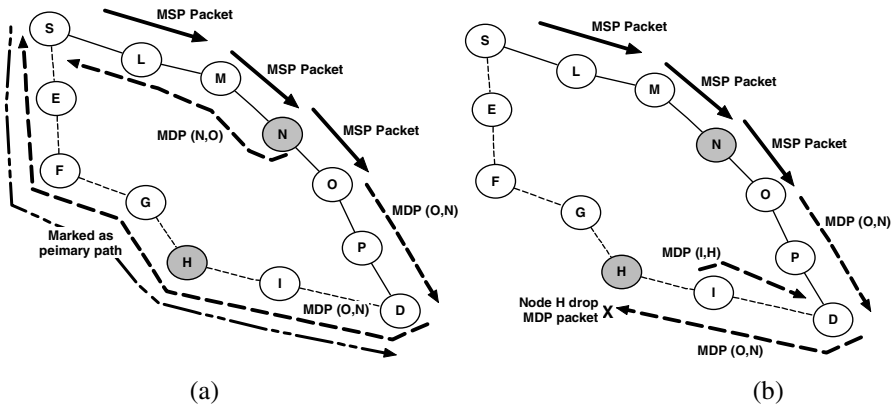


Fig. 5. The source node S (a) receives the MDP packet (b) don't receive the MDP packet

3.2.4 Misbehaving Isolation Phase

When a neighbor of a malicious node detect its malicious activity it will send the MDP packet and put the malicious node ID on its black list to isolate it. Also when the trust value of a given node reduced below a given threshold it will be marked as misbehaving node and its ID inserted in the black list.

After small number of transaction all malicious node's neighbors will put its ID on their black lists, so the malicious node will be fully isolated from MANET. The misbehaving node can rejoin the network only if it moves from its location and have new neighbors and its trust value is above the threshold.

3.2.5 Route Management Phase

When an intermediate node detects a link failure due to Movement, down of nodes or the apparent of a selfish node lying along an active path, it generates a RERR packet. The RERR packet propagates toward the traffic source having a route via the failed

link, and erases all broken routes on the way. When the source upon receive the RERR packet it initiates a new route discovery if it still needs the route.

4 Conclusion and Future work

In this paper we proposed a general solution to packet dropping misbehavior in MANET. The solution allows monitoring, detecting, and isolating the malicious node. In TRIDNT the malicious node neighbors will isolate it after a few numbers of transactions. Also TRIDNT allow a controlled amount of node selfishness behavior to give an incentive to the selfish nodes to declare its selfishness behavior to its neighbors, which reduce the searching time of misbehaving nodes to search for malicious nodes only. Then we can find an isolate the malicious node; denied access to the network; in small amount of time without using promiscuous listening, which resulting in an improved overall throughput performance for MANET.

In the future we will design a trust model to calculate the node and path trust values, and define a trustworthy accurate threshold. Also we will simulate TRIDNT to show the results and effectiveness of our solution, and compare it with existing trust based routing algorithms like TWOACK, and Muhammad Zeshan proposed schemes. A detailed simulation evaluation will be conducted in terms of Routing Packet Overhead, Security Analysis, Mean Time to detect dropper node, Overall Network Throughput, and Average Latency. Also we will study the situation when there are more than one malicious node in the route from the source and destination.

References

1. Johnson, D., Maltz, D.: Dynamic Source Routing protocol (DSR) for Mobile Ad Hoc Wireless Networks for IPv4. RFC 4728 (February 2007), <http://www.ietf.org/rfc/rfc4728.txt>
2. Perkins, C.E., Belding-Royer, E., Das, S.R.: Ad hoc on-demand distance vector (AODV) routing, RFC 3561 txt (July 2003), <http://www.ietf.org/rfc/rfc3561>
3. Perkins, C.E., Bhagwat, P.: Highly Dynamic Destination- Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In: Proc. ACM SIGCOMM, pp. 234–244 (1994)
4. Vaidya, B., Yeo, S.S., Choi, D.-Y., Jo Han, S.: Robust and secure routing scheme for wireless multihop network. Personal and Ubiquitous Computing magazine (April 4, 2009)
5. Hu, Y.-C., Perrig, A., Johnson, D.: Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks. Wireless Networks Journal 11, 21–38 (2005)
6. Sanzgiri, K., LaFlamme, D., Dahill, B., Neil Levine, B., Shields, C., Belding-Royer, E.M.: Authenticated Routing for Ad Hoc Networks. Proceedings of IEEE journal on selected areas in communications 23(3) (March 2005)
7. Mahapatra, R.P., SM IACSIT, Katyal, M.: Taxonomy of Routing Security for Ad-Hoc Network. International Journal of Computer Theory and Engineering 2(2), 303–307 (2010)
8. Wang, D., Hu, M., Zhi, H.: A Survey of Secure Routing in Ad Hoc Networks. In: The Ninth International Conference on Web-Age Information Management (waim), pp. 482–486 (2008)
9. Hu, Y.-C., Perrig, A.: A Survey of Secure Wireless Ad Hoc Routing. IEEE Security and Privacy 2(3), 28–39 (2004)

10. Gong, W., You, Z., Chen, D., Zhao, X., Gu, M., Lam, K.-Y.: Trust Based Routing for Misbehavior Detection in Ad Hoc Networks. *Journal Of Networks* 5(5), 551–558 (2010)
11. Azer, M.A., El-Kassas, S.M., Hassan, A.W.F., El-Soudani, M.S.: A Survey on Trust and Reputation Schemes in Ad Hoc Networks. In: *Third International Conference on Availability, Reliability and Security*, pp. 881–886 (2008)
12. Aad, I., Hubaux, J.-P., Knightly, E.W.: Impact of Denial of Service Attacks on Ad Hoc Networks. *IEEE/ACM Transactions on Networking* 16(4), 791–802 (2008)
13. Aad, I., Hubaux, J.-P., Knightly, E.W.: Denial of service resilience in ad hoc networks. In: *Proceedings of Mobicom* (2004)
14. Kuzmanovic, A., Knightly, E.: Low-rate TCP-targeted denial of service attacks (the shrew vs. the mice and elephants). In: *Proceedings of ACM SIGCOMM* (2003)
15. Marti, S., Giuli, T.J., Lai, K., Baker, M.: Mitigating routing misbehavior in mobile ad hoc networks. In: *Proceedings of the Sixth annual ACM/IEEE International Conference on Mobile Computing and Networking*, pp. 255–265 (2000)
16. Buchegger, S., Le Boudec, J.-Y.: Performance Analysis of the CONFIDANT Protocol (Cooperation Of Nodes: Fairness In Dynamic Ad-hoc Networks). In: *Proceedings of the 3rd ACM International Symposium on Mobile and Ad Hoc Networking & Computing (MobiHoc 2002)*, Lausanne, Switzerland, pp. 226–236 (June 2002)
17. Balakrishnan, K., Deng, J., Varshney, P.K.: Twoack: preventing selfishness in mobile ad hoc networks. In: *The IEEE Wireless Communication and Networking Conference(WCNC 2005)*, New Orleans, LA,USA, pp. 2137–2142 (March 2005)
18. Zeshan, M., Khan, S.A., Cheema, A.R., Ahmed, A.: Adding Security against Packet Dropping Attack in Mobile Ad Hoc Networks. In: *International Seminar on Future Information Technology and Management Engineering*, pp. 568–572 (2008)
19. Marina, M.K., Das, S.R.: Ad hoc on-demand multipath distance vector routing. *Wirel. Commun. Mob. Comput.*, 969–988 (2006), <http://www.interscience.wiley.com>
20. Nasipuri, A., Das, S.R.: On-demand multipath routing for mobile ad hoc networks. In: *Computer Eight International Conference on Communications and Networks*, pp. 64–70 (1999)

Physiologically Based Speech Synthesis Using Digital Waveguide Model

A.R. Patil and V.T. Chavan

DKTE's Textile and engineering institute, Ichalkaranji
anjalirpatil@gmail.com, vandana.t.chavan@gmail.com

Abstract. Problem in the area of speech synthesis research is that synthetic speech generally sounds unnatural. Artificially generated speech lacks the smoothness of natural speech, and this in turn can hinder intelligibility. Natural sounding synthetic speech can be generated by modeling inflection, modeling the glottal source, improving techniques of concatenating segments of speech, and modeling the temporal nuances of phrasing. Computer models of the human voice-production system are an important research tool for speech scientists and engineers. This paper uses physiological data to model human vocal tract to tackle above mentioned problem. Main focus is on piece-wise cylindrical model of the human vocal tract, since it is a relatively simple (in theory), and is capable of producing good results and most natural sounding speech synthesis. And it attempts to describe the comparison between different numbers of segments used to model vocal tract.

General terms: Articulatory speech synthesis, digital waveguide modeling, vocal tract.

Keywords: formant frequency, cylindrical segments, DWG, scattering.

1 Introduction

There are currently a number of different approaches used to model human speech, some of which are based on a signal models (format synthesis), and some of which are based on physical models, finite element ,piece-wise conical/cylindrical segments (Concatenate synthesis) and some are based on computational models of the human vocal tract and the articulation processes occurring there.(Articulatory synthesis).During speech production, coordinated movements of the tongue, jaw, lips, and to some degree the larynx, continuously alter the shape of the vocal tract i.e. pharynx and oral cavity. Movement of the soft palate varies the acoustic coupling of the vocal tract to the nasal passages, and also may slightly change the shape of the upper pharynx. Integrated actions of individual articulators facilitate the creation of time-varying acoustic resonances that transform the sound generated by vocal-fold vibration or turbulence, into the stream of vowels and consonants that comprises speech. Specifically, it is the articulators' collective effect on the variation in cross-sectional area along the length of the vocal tract i.e., the *area function* and coupling to the nasal tract, as well as other possible side branch cavities, that is most closely related to the

pattern of acoustic characteristics expressed in the speech waveform. Hence, a simplified view of speech production may consist of a tubular system whose cross-sectional area varies, as a function of time, emulates that of a real vocal tract. This view forms the basis of a certain class of speech production models that operate on a parametric representation of the vocal-tract area function, and allow for calculation of corresponding acoustic characteristics. Area function models contrast with “articulatory” models in which the positions of individual articulators or some form of vocal-tract shaping components are represented in the midsagittal plane. These are intuitively appealing because of the physiological correlation between model parameters and human articulatory structures, and their ability to replicate observed articulatory movement. As a result, articulatory-type models are well suited for investigating and establishing speech motor control strategies. The relation of the articulatory parameters to the acoustic characteristics is, however, typically mediated by an empirically based conversion of midsagittal cross dimensions to the area function. Hence, control of the detailed vocal-tract shape at the level of cross-sectional area is less direct than with an area function model. A possible exception is a recently developed model that utilizes midsagittally based control parameters, but avoids the cross-dimension transformation to area by generating vocal-tract shapes based on data obtained from MRI.

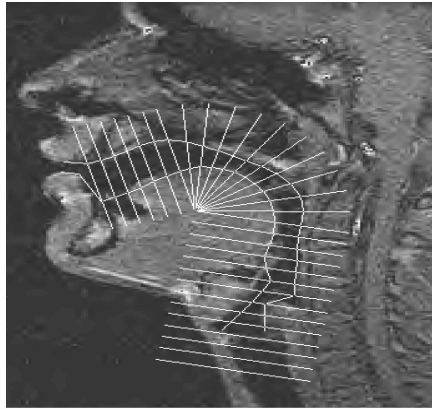


Fig. 1. MRI cross-section of vocal tract

2 Relevant Work

The most straightforward form of an area function model consists of a direct specification of the cross-sectional areas extending from the glottis to the lips. The parameters in this case are simply the areas themselves. Variation over time requires interpolation from one complete area function (representing one phonetic element) to another. In this approach, area data obtained from imaging studies[4] can be used directly, but the ability to create realistic time-varying vocal-tract shapes (i.e. area functions that did not exist in the original data set) is limited. Specification of the area function with a small

set of physiologically relevant parameters forms the basis for more parsimonious models. Examples are the well-known “three parameter” models [6], where the constriction location X_c (distance from glottis or lips to the constriction) and area A_c are specified along with a ratio of the length of the lip opening to its area (l/A). The areas corresponding to the tongue section are determined by a continuous mathematical function (e.g. parabola). To be more flexible in the variety of shapes that can be generated, these models have been modified in various ways. Extended the number of parameters to five, whereas Lin 1990 Incorporated separate continuous functions for the back and front cavities. Another type of area function model was proposed by [8], where the parameters were derived purely from acoustic considerations. The vocal tract was divided into separate distinct regions, each of which has sensitivity to formant frequency change that is predictably related to an increase or decrease in cross sectional area of a particular region. To control the first three formant frequencies, the cross-sectional area of eight regions of unequal length must be specified as parameters. This model is perhaps less interpretable than the previous ones in terms of articulation, but is interesting in the sense that sufficient control parameters could be derived in the absence of articulatory knowledge. An eventual goal of developing a parametric area function model is to accurately reproduce connected speech. That is, speech created by a vocal tract whose shape alternates between those of *vowels* and consonants or from one *vowel* to another. Whereas the models discussed previously are most relevant for *vowel* articulations consonant characteristics are not specifically parameterized, it is conceivable that they could be modified or extended to create consonant-like *voc[1]* perhaps by allowing the minimum area to approach or become zero. Simulation of connected speech would then be carried out by interpolating a sequence of parameter values over the time course of an utterance. A linear sequencing of vowel and consonant events, however, is limited in its representation of co articulation.

3 Experimental Setup

For creating a vocal-tract model, the continuous cross-sectional area function of the airway (which may be obtained by magnetic resonance imaging or X-ray micro beam tracking), is sampled for digital simulation such that the tract consists of many cylindrical segments of equal lengths but different cross-sectional areas. The cross-sectional area of any cylinder can be independently varied to simulate the changing shape of the vocal tract during articulation.

Fig. 2 shows a plot of the cross-sectional areas of the segments versus segment number, for a *vowel li*.

The series of cylindrical tubes described above can be modeled as a Digital Wave Guide (DWG). A DWG uses delay (memory) elements to represent the spatial position of forward and backward traveling waves in the physical system. The digital wave guide for a cylindrical tube is shown in figure 3. The number of delay elements, N , in the wave guide depend on *a*) the length of the tube, *b*) the velocity of sound, and *c*) the spatial sampling frequency according to:

$$d=c/2 F_s$$

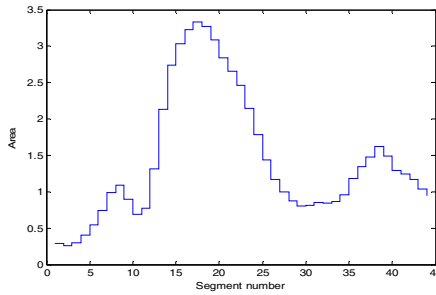


Fig. 2. Cross-sectional areas

For example, with spatial sampling frequency $F_s = 44100\text{Hz}$, and given that $c = 350\text{m/s}$, will require $d = .39\text{cm}$. Considering a male vocal tract of about 17.5 cm we can divide the total tract into 44 segments.

Digital waveguide modeling is a computational physical modeling technique that is widely used in the field of speech synthesis, acoustics and computer music. In this method, solids or cavities where wave propagation can be assumed to be linear and one-dimensional are modeled as waveguides. They are discretized in space and time, so that they can be conveniently represented on a computer. The waveguide models consist of delay lines, digital filters and non-linear elements.

The air cavity in the human vocal tract acts as a waveguide for pressure (or volume velocity) waves that emanate from the glottis.

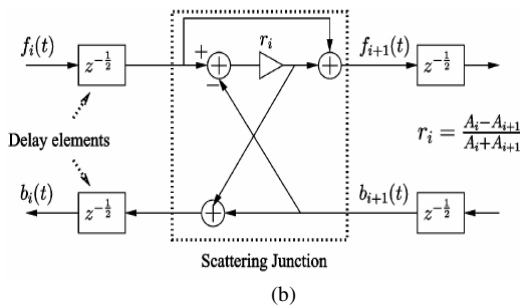
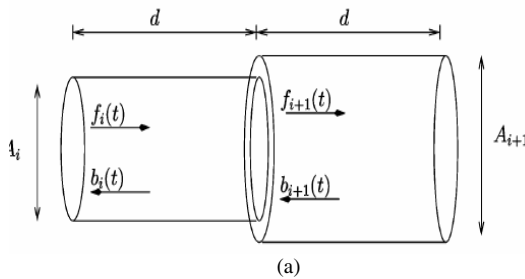


Fig. 3. Waveguide model for a junction of two cylindrical segments

Above fig. 3 shows the junction of any two cylindrical segments. The lengths of the tubes can be mapped to delays in a digital waveguide, while the ratio of cross-sectional areas can define the reflection coefficient r_i .

$$r_i = \frac{A_i - A_{i+1}}{A_i + A_{i+1}}$$

The pressure wave in the i^{th} segment, at any given instant, is the sum of the forward propagating component and the back ward propagating component, [Fig. 3(a)]. The characteristic impedance encountered by a wave in a cylindrical tube of cross-sectional area is,

$$Z = \rho c / A$$

where ρ is the density of air.

Fig. 3 shows the complete waveguide model for a 44-segment vocal tract. The glottal end of the tube is modeled as closed and the lip end as open, with reflection coefficients and respectively. A voice source model [11] produces a time-varying volume velocity wave, which is input to the vocal tract filter. The output speech is the sum of component in the upper delay line and component in the lower delay line, at the end(lip) segment.

This paper implements a DWG model of the human vocal tract. Although the results are far from perfect, it has shown how it is possible to synthesize speech using a simple piece-wise cylindrical model of the vocal tract along with 8 tube model, 35 tube and 44 tube model.

The Matlab implementation of the piece-wise cylindrical DWG model was created in order to demonstrate the feasibility of this approach and in order to experiment with the model. For the comparison of this different number tube in this paper the data is taken for 8-segments, 35 segments and for 44-segments for the vowel /i/.

This data can be obtained using X-ray or MRI data; however, since the vocal tract is a time-varying filter, it is difficult to obtain data on the temporal aspects of speech, especially plosives and fricatives. Also, due to the variability between subjects (gender, age, background, etc), it is impossible to obtain exhaustive model data.

This paper also gives voiced/unvoiced control, to control whether speech is noisy (consonants) or voiced (vowels). When the model is set to UNVOICED filtered noise is injected at some point along the length of the wave guide (location depends on consonant being synthesized).

It was difficult to find shape data to control this model we looked into Perry Cook's SPASM model [5] and collected data for the 8-tube model. The data is collected for 35 and 44 segments from Story 1995.

4 Results

Proposed Computational model of the human vocal tract is hybrid approach of 8-segments, 35 segments and for 44-segments. Hence the results has to be given stepwise:

Firstly Ideal formant frequency for /i/ - vowel is given in table 1 below. The presumption is that if tubes were made with these area functions and were excited with a source with a rich harmonic content like the vocal folds, they would produce a sound similar to the vowel sounds. This has been demonstrated in this paper.

Each original area function has been segmented to consist of either 8,35,44 area sections given in cm^2 .

Fig 2 represents the 44-cross-sectional areas for the vowel /i/ by dividing the complete tract into 44 segments. While as fig. 4 below shows the 35- cross-sectional areas for the same vowel.

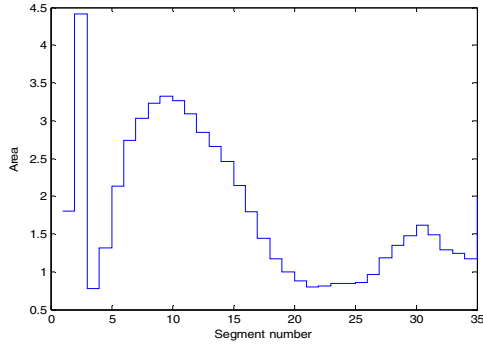


Fig. 4. Cross-sectional areas for the 35 equal-length cylindrical segments in /i/vowel

The figure below [fig.5] shows area for the same vowel for 8-segments.

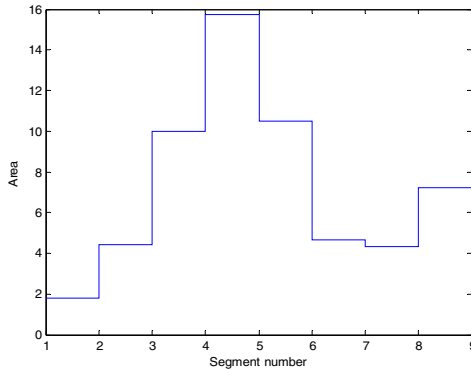


Fig. 5. Cross-sectional areas for the 8 equal-length cylindrical segments in /i/vowel

The same algorithm is used to generate the sound for different segments. Results for the 44 segment the formant frequency is observed as shown in fig 6.in magenta color The figure shows that results are satisfactory compared with the ideal frequency shown in table 2.Similar results can be obtained for remaining *vowels* and consonants. After testing, results found were satisfactory.

It is observed that for the 35 segments the results obtained are shown in red color in fig.8 closer to the ideal Formant values.

And that for the 8-segment it is far away from the natural.(in blue color)

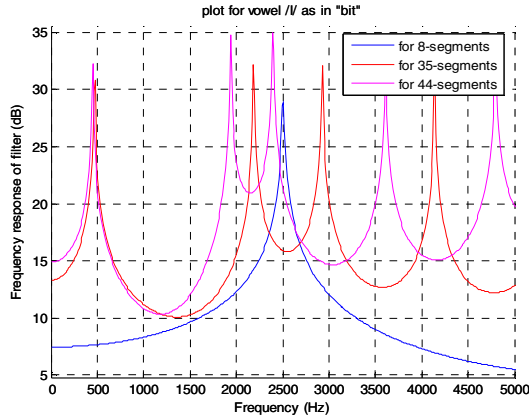


Fig. 6. Formant plot For 8 ,35,44 segments for vowel /i/

Table 1. Formant frequencies for different number of segment

/i/	F1	F2	F3
Standard	400HZ	1900HZ	2550HZ
44-segments	400HZ	1900HZ	2400HZ
35-segments	450HZ	2200HZ	2800HZ
8-segments	?	?	2500HZ

5 Conclusions

The resulting formant frequencies were found to be in close agreement with their expected values. The algorithms are computationally efficient as they require a relatively small overhead (approximately 5%) for increased segments. The system sampling frequency stays constant even as the total length of the tract is varied, thus eliminating the need for re-sampling of the synthesized speech.

For this work it is implemented as 8-tube,35-tube,44-tube model of the vocal tract. In order to improve the accuracy of the model more tubes should be used (better approximation of the vocal tract shape), and a variable length could be implemented using fractional delays (more realistic, eg. lip elongation). A better model of the glottal excitation produces more natural sounding speech. Though the results are still not perfect natural one can achieve best by increasing the number of tubes. Again more accurate results can be obtained by changing the algorithm for fractional elongation of particular segments or by lip lengthening and larynx lowering. Actually this is tested only for *vowels*; Future work could include synthesizing more complex words and sentences from their physiological models.

Acknowledgements

Authors thanks to Prof. D.V.Kodawade, head of computer science & engineering dept. ,Textile and Engineering Institute, Ichalkaranji, and Prof. Mrs. L.S.Admuthé, vice principal, Textile and Engineering Institute, Ichalkaranji for their helpful comments and discussion. We also thank Textile and Engineering Institute, Ichalkaranji (M.S. INDIA) for providing platform.

References

- [1] Valimiaki, V., Rabenstein, R., Rocchesso, D., Serra, X., Smith, J.O.: Signal Processing for Sound Synthesis: Computer-Generated Sounds and Music for All. *IEEE Signal Processing Magazine* (2007)
- [2] Closed-Form Design of Half-Sample Delay IIR Filter Using Continued Fraction Expansion. Dept of comp. Commun. Eng., Nat. Kaohsiung First Univ. of Sci. & Technol. *Circuits and Systems I: Regular Papers, IEEE Transactions on [Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on]* (March 2007)
- [3] Mathur, S., Story, B.H., Rodriguez, J.J.: Vocal-Tract Modeling: Fractional Elongation of Segment Lengths in a Waveguide Model with Half-Sample Delays. *IEEE Transactions on Audio, Speech, and Language processing* (September 2006)
- [4] A parametric model of the vocal tract area function for vowel and consonant simulation-Brad H. Story. *Speech Acoustics Laboratory, Department of Speech and Hearing Sciences, University of Arizona, Tucson, Arizona 85721*
- [5] Valimiaki, V.: Discrete-time modeling of acoustic tubes using fractional delay filters. Ph.D. dissertation, Helsinki Univ. Technol., Lab. Acoust. Audio Signal Process., Espoo, Finland (1995)
- [6] Cook, P.R.: SPASM, a Real-Time Vocal Tract Physical Model Controller; and Singer, the Companion Software Synthesis System. *Computer Music Journal* 17(1) (1993)
- [7] Mullen, J., Howard, D.M., Murphy, D.T.: Waveguide physical modeling of vocal tract acoustics: flexible formant bandwidth control from increased model dimensionality. *IEEE Transactions, Audio, Speech and Language Processing Media Eng. Group, York Univ. UK.* (May 2006)
- [8] Carré, R., Divenyi, P.L.: Modeling and Perception of 'Gesture Reduction'. *Phonetica- international journal of phonetic science* 57(2-4) (2000)
- [9] Mathur, S., Story, B.H.: Vocal tract modeling: implementation of continuous length variations in a half-sample delay Kelly-Lochbaum model. In: *Proceedings of the 3rd IEEE International Symposium on Signal Processing and Information Technology, ISSPIT 2003, Electr. & Comput. Eng., Arizona Univ., Tucson, AZ, USA, December 14-17 (2003)*
- [10] Deller Jr, J.R., Hansen, J.H.L., Proakis, J.G.: *Discrete-time Processing of Speech Signals.* IEEE Press, Piscataway (2000)
- [11] Kelly, J.L., Lochbaum, C.C.: Singing Kelly-Lochbaum Vocal Tract in published a software version of a digitized vocal-tract analog mode (1962)
- [12] Sondhi, M.M., Schroeter, J.: A hybrid time-frequency domain articulatory Speech synthesizer. *IEEE Trans. Signal Processing* 35(7), 955-967 (1987)

Improving the Performance of Color Image Watermarking Using Contourlet Transform

Dinesh Kumar¹ and Vijay Kumar²

¹ CSE Department, GJUS&T, Hisar, Haryana, India

² CSE Department, JCDMCOE, Sirsa, Haryana, India

dinesh_chutani@yahoo.com, vijaykumarchahar@gmail.com

Abstract. In this paper, a color image watermarking scheme based on combination of discrete cosine transform, contourlet transform and singular value decomposition has been proposed. The color image is transformed into three color components luminance (Y) and chrominance (Cb and Cr). After that, discrete cosine transform is applied on the components to obtain DCT coefficients. The DCT coefficients thus obtained are decomposed into directional sub bands using 2-level contourlet transform. A grey scale watermark is decomposed using contourlet transform. Laplacian Pyramid decomposition is applied on both, fourth directional subband from watermark and cover image to obtain Low pass subband. The singular values of lowpass subband of watermark are embedded into singular values of lowpass subband of cover image. The experimental results demonstrate that the proposed watermarking scheme is robust against common image processing attacks. The comparison analysis reveals that the proposed watermarking scheme outperforms the color image watermarking scheme reported recently.

Keywords: Contourlet Transform, Discrete Cosine Transform.

1 Introduction

The recent years have witnessed the development of efficient copyright protection techniques to cope up the urgent need in the multimedia companies due to illegal manipulation and reproduction of digital data. Digital watermarking is the one among a wide variety of techniques proposed so far for copyright protection of digital data that is most frequently used due to its potential applications such as in medical imaging, digital camera and video-on-demand systems etc. [1].

In this technique, watermark is embedded into digital data that can be extracted later in order to claim the ownership of data. Digital watermarking techniques can be broadly categorized into two groups, the spatial domain and frequency domain watermarking techniques [2, 4, 5]. Spatial domain watermarking techniques embed watermarks in spatial characteristics of cover image directly. They are easy and fast, but less resistant against noise attacks. Frequency domain watermarking techniques transform the cover image into frequency domain coefficients before embedding watermarks in it. These techniques are complex, but more effective against attacks.

Commonly used frequency domain watermarking techniques include the Discrete Wavelet Transform (DWT), Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), Discrete Hadamard Transform (DHT) and Singular Value Decomposition (SVD). Of these, DWT and the SVD methods are known to have gained much more popularity. But, these techniques lack in capturing the directional information such as directional edges of the image. This problem can be solved by Discrete Contourlet Transform. Discrete Contourlet Transform (CT) is capable of capturing the directional information with multiresolution representation [6, 12]. It makes use of Laplacian Pyramid for multiresolution representation of the image [9, 10] followed by a directional decomposition on every band pass image using directional filters [8, 10]. In Ref. [7], authors use the approach of Quantization Index Modulation and multiple Descriptions on gray scale host image and binary watermark.

In this paper, a color image watermarking scheme using CT has been proposed to improve the performance of watermarking scheme. First, the cover image is transformed from RGB Color Space to YCbCr Space. After that, DCT is applied on each components of the color image. CT is performed on DCT coefficients. The singular values of all directional sub bands of YCbCr are modified with singular values of watermark.

The remainder of this paper is organized as follows. Section 2 briefly explains the background relevant to proposed watermarking scheme. Section 3 introduces the proposed color image watermarking scheme. Section 4 shows the experimental results followed by conclusions in section 5.

2 Background

There are three techniques that are used in proposed color image watermarking scheme, namely the DCT, SVD and CT.

The SVD technique has been applied to large number of applications such as pattern recognition and signal processing [2, 15]. The SVD of an $K \times M$ image matrix \mathbf{X} is a decomposition of the form

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T \tag{1}$$

where matrices \mathbf{U} and \mathbf{V} are $K \times K$ and $M \times M$ orthogonal matrices respectively. \mathbf{S} is a $K \times M$ diagonal matrix with singular values (SV) on the diagonal. The main property of SVD relevant to watermarking is that larger singular values of an image don't change significantly when image processing attacks are performed on the image [2].

The DCT is a technique for converting a signal into elementary frequency components [2, 3]. It is applied on an image \mathbf{I} , having $K \times M$ pixels to transform the image according to equation 2 [2]:

$$y(u, v) = \sqrt{\frac{2}{M}} \sqrt{\frac{2}{K}} \alpha_u \alpha_v \sum_{u=0}^{M-1} \sum_{v=0}^{K-1} I(m, k) \cos\left(\frac{(2m+1)u\pi}{2M}\right) \cos\left(\frac{(2k+1)v\pi}{2k}\right) \tag{2}$$

where $y(u, v)$ is the DCT coefficients in row u and column v of the image matrix and $I(m, k)$ is the intensity of the pixel in row m and column k of the image. The values of α_u and α_v are both set to $\frac{1}{\sqrt{2}}$ when $u, v = 0$, Otherwise 1.

The image can be reconstructed by applying IDCT according to equation 3 [2]:

$$I(m, k) = \sqrt{\frac{2}{M}} \sqrt{\frac{2}{K}} \sum_{u=0}^{M-1} \sum_{v=0}^{K-1} \alpha_u \alpha_v y(u, v) \cos \frac{(2m+1)u\pi}{2M} \cos \frac{(2k+1)v\pi}{2k} \quad (3)$$

The CT was proposed by M. N. Do and Martin Vetterlin[6, 11]. It is constructed by two filter-bank stages, a Laplacian Pyramid (LP) followed by a Directional Filter Bank (DFB) [12]. The LP decomposes the image into frequency band to obtain singular points. The DFB decomposes each LP detail band to capture directionality.

3 Proposed Color Image Watermarking Scheme

Color Image Watermarking Scheme proposed by Rawat and Raman [14] includes the transformation of color cover into YCbCr color space. First DCT was applied on three components of the cover image and the components were further decomposed using wavelet packet transform. The watermark image was embedded into all the frequency components of the image.

In this paper, instead of using wavelet packet transform, we have used CT to capture the directional information also.

3.1 Embedding Procedure

The steps of embedding procedure are as follows:

1. Apply 2-level CT on the watermark to obtain directional sub bands.
2. The fourth directional subband is decomposed using LP decomposition.
3. Perform SVD on bandpass image obtained from Step2.

$$\mathbf{W} = \mathbf{U}_w \times \mathbf{S}_w \times \mathbf{V}_w^T \quad (4)$$

where \mathbf{W} is the bandpass image of watermark.

4. Transform the cover image from RGB color space into YcbCr color space.
5. For each color component of color space
 - (a) Perform DCT.
 - (b) Perform 2-level CT to obtained directional sub bands.
 - (c) Perform LP decomposition on fourth directional subband.
 - (d) Perform SVD on bandpass image obtained from Step 5-(c).

$$\mathbf{CBP}_i = \mathbf{UBP}_i \times \mathbf{SBP}_i \times \mathbf{VBP}_i^T \tag{5}$$

where \mathbf{CBP}_i represent the fourth direction subband of cover image.

- (e) Modify the singular values of bandpass image.

$$\mathbf{SBP}'_i = \mathbf{SBP}_i + \alpha \times \mathbf{S}_w \tag{6}$$

where \mathbf{SBP}'_i is modified singular matrix, \mathbf{SBP}_i is singular matrix of directional subband of cover image, \mathbf{S}_w is singular matrix of watermark and α is the amplification factor.

- (f) Obtain the modified band pass image.

$$\mathbf{CBP}^*_i = \mathbf{UBP}_i \times \mathbf{SBP}'_i \times \mathbf{VBP}_i^T \tag{7}$$

where \mathbf{CBP}^*_i is modified band pass of cover image, \mathbf{UBP}_i and \mathbf{VBP}_i are orthogonal matrices of band pass of cover image.

- (g) Apply LP reconstruction on modified bandpass image.
 - (h) Perform inverse CT (ICT).
 - (i) Perform IDCT.
6. Transform the image into RGB color space to get the watermarked image.

3.2 Extraction Procedure

The steps of extraction procedure are as follows:

1. Transform the cover image and watermarked image from RGB color space into YCbCr Space.
2. For each color component of cover and watermarked images
 - (a) Perform DCT.
 - (b) Perform 2-level CT to obtained directional sub bands.
 - (c) Perform LP decomposition on fourth directional subband.
 - (d) Perform SVD on bandpass image of cover and watermarked image.

$$\begin{aligned} \mathbf{CBW}_{wi} &= \mathbf{UBW}_{wi} \times \mathbf{SBW}_{wi} \times \mathbf{VBW}_{wi}^T \\ \mathbf{CBP}_i &= \mathbf{UBP}_i \times \mathbf{SBP}_i \times \mathbf{VBP}_i^T \end{aligned} \tag{8}$$

where \mathbf{CBW}_{wi} and \mathbf{CBP}_i are bandpass of watermarked and cover image.

- (e) Extract singular values of watermark as

$$\mathbf{S}_w^{ext} = \frac{\mathbf{SBW}_{wi} - \mathbf{SBP}_i}{\alpha} \tag{9}$$

where \mathbf{SBW}_{wi} and \mathbf{SBP}_i are singular matrices of watermarked and cover image respectively. \mathbf{S}_w^{ext} is extracted singular matrix of watermark.

3. Perform SVD on all bandpass of watermark

$$\mathbf{W}^{ext} = \mathbf{U}_w \times \mathbf{S}_w^{ext} \times \mathbf{V}_w^T \quad (10)$$

where W^{ext} is extracted watermark image.

4. Apply LP reconstruction on results obtained from Step 3.
5. Perform ICT to get the watermarks.

4 Experimentation and Results

4.1 Experiment 1 and Results

Our experimental work considers six cover images: Goldhill, Redfort, Lena, Nuclear Plant, Boat and Barbara, each of size 256×256 , and one watermark gray scale image named Springer logo of size 90×256 . The CT uses ‘pkva’ Pyramid filter and ‘pkva’ Directional filter to obtain a two level decomposition. Amplification factor is set to 0.01 for embedding watermark. The Springer logo is embedded into all the cover images. Figure 1 shows watermarked images and extracted watermark image using proposed method.

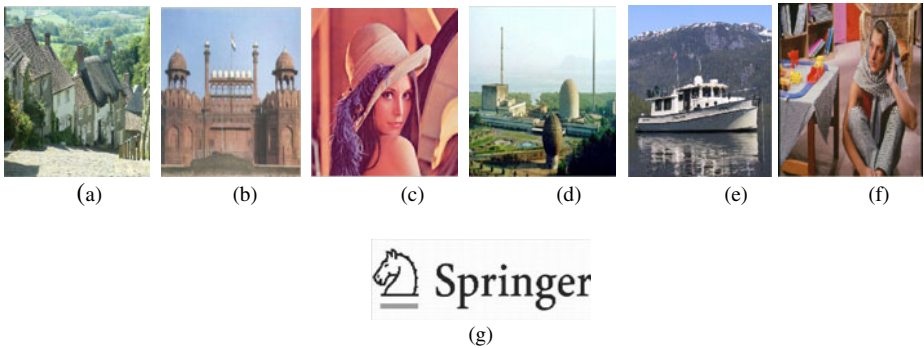


Fig. 1. Watermarked images; (a) Goldhill (b) Redfort (c) Lena (d) Nuclear Plant (e) Boat (f) Barbara and Extracted watermark image; (g) Springer Logo

Table 1. Quality metrics of watermarked images using Proposed Method

	Lena	Barbara	Goldhill	Redfort	Boat	Nuclear
PSNR	43.2226	42.6960	43.1691	43.1228	43.0486	43.7958
SSIM	0.9845	0.9947	0.9976	0.9875	0.9915	0.9815

The Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index (SSIM) [13] values of each watermarked image are tabulated in table 1.

Tables 2 and 3 show the comparison between proposed and Rawat and Raman’s method [14] in terms of both quality of watermarked image and extracted watermark image. PSNR values of watermarked images using proposed method are better than

Table 2. Comparison of watermarked images in terms of PSNR

	Lena	Barbara	Goldhill	Redfort	Boat	Nuclear
Rawat and Raman Method [14]	34.1231	32.3770	32.0585	32.0845	32.0160	32.6764
Proposed Method	43.2226	42.6960	43.1691	43.1228	43.0486	43.7958

Table 3. Comparison of extracted watermark images in terms of correlation coefficients

	Lena	Barbara	Goldhill	Redfort	Boat	Nuclear
Rawat and Raman Method [14]	0.8244	0.8456	0.8333	0.8385	0.8096	0.8457
Proposed Method	0.9940	0.9941	0.9936	0.9932	0.9930	0.9942

the latter one. Similarly, the extracted watermark from all the watermarked images gives better correlation coefficients.

4.2 Experiment 2 and Results

The next experiment was performed to see the effect of chosen attacks such as Sharpening, Gamma Correction, Affine Transform, Histogram Equalization, Resizing, Cropping, Rotation and Gaussian blur. Figure 2 shows the Nuclear Plant watermarked images after applying above said attacks. The best extracted watermarks are also shown in figure 2.

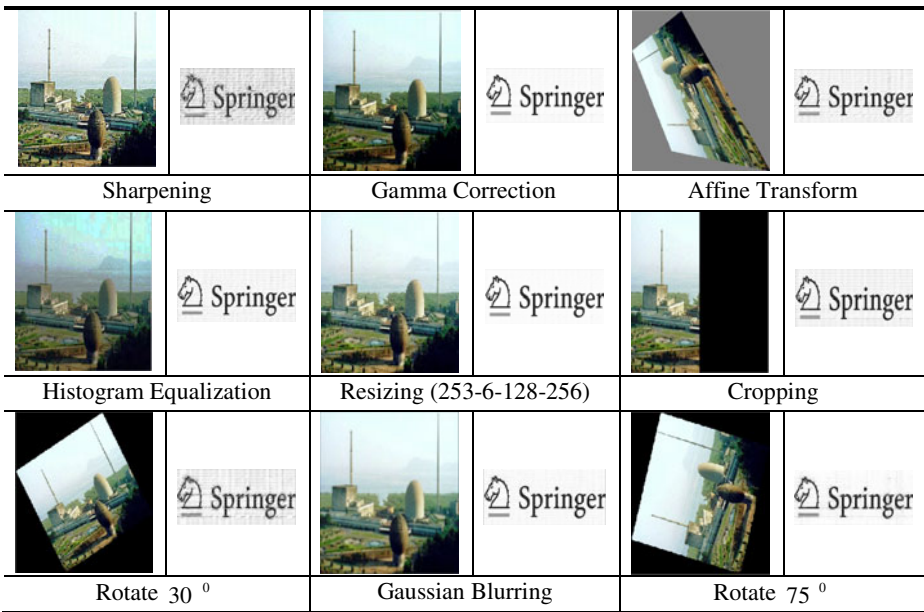


Fig. 2. Extracted Springer logo from watermarked image nuclear Plant after various image Processing attacks

Table 4. Correlation coefficients of the best extracted watermark images under different image processing attacks

Attacks	Watermarked Images					
	Lena	Barbara	Goldhill	Redfort	Boat	Nuclear
Sharpening	0.4247	0.4749	0.6266	0.6414	0.7124	0.6066
Gamma Correction	0.8522	0.9789	0.9561	0.9726	0.9908	0.9859
Transform	0.6232	0.5902	0.9557	0.6090	0.9605	0.8929
Hist. Equalization	0.6384	0.7095	0.9670	0.5856	0.8407	0.9501
Resizing	0.9864	0.9897	0.9956	0.9957	0.9955	0.9948
Cropping(right half)	0.5178	0.7299	0.9879	0.7478	0.9869	0.9307
Cropping(right half)	0.7402	0.5003	0.9284	0.6611	0.9682	0.6058
Rotate 30 ^o	0.4575	0.4134	0.8622	0.4334	0.9283	0.7203
Rotate 75 ^o	0.6164	0.5623	0.8964	0.5503	0.7832	0.8701
Gaussian Blurring	0.9232	0.9086	0.9717	0.9882	0.9826	0.9735

The correlation coefficients of best extracted watermark image after different image processing attacks are illustrated in table 4. The results reveal that extracted watermark images have good correlation coefficients.

Next, we saw the effect of adding different types of noises on watermarked images. Figure 3 shows the extracted watermark and watermarked image after adding different types of noise attacks. Table 5 depicts the correlation coefficients of best extracted watermark image. The results reveal that extracted watermark has better correlation coefficients and good visual quality under different noise attacks.

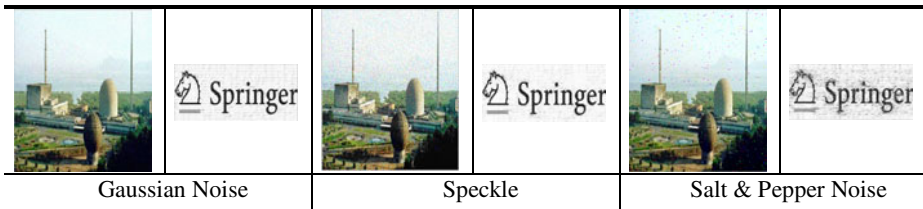


Fig. 3. Extracted Springer logo from watermarked image nuclear Plant after adding different types of noises

Table 5. Correlation coefficients of the best extracted watermark images under different types of noises

Attacks	Watermarked Images					
	Lena	Barbara	Goldhill	Redfort	Boat	Nuclear
Salt & Pepper (0.01)	0.7148	0.8125	0.7888	0.6984	0.7582	0.6715
Gaussian Noise (0.01)	0.9063	0.9285	0.9703	0.8691	0.9294	0.9204
Speckle (0.01)	0.7693	0.8978	0.8563	0.6969	0.8891	0.9316

Table 6. Quality metrics of the watermarked images under different scaling factor using Nuclear Plant cover image and Springer Logo image

	Scaling Factor									
	0.01	0.03	0.05	0.07	0.09	0.1	0.3	0.5	0.7	0.9
PSNR	43.79	43.79	43.72	43.56	43.28	43.11	38.19	34.26	31.43	29.28
SSIM	0.982	0.982	0.981	0.981	0.981	0.981	0.968	0.952	0.933	0.915

We further performed the simulation to see the effect of scaling factor on watermarked image. Table 6 gives PSNR and SSIM for different values of scaling factor on watermarked image. A careful look at the table 6 reveals that the increase in value of scaling factor results in decrease in values of both PSNR and SSIM.

5 Conclusion

This paper has presented a color image watermarking scheme for embedding a gray scale watermark into color cover image using CT together with DCT and SVD. The watermarked images were subjected to common image processing attacks. The experimental results depict that the proposed watermarking scheme outperforms Rawat and Raman’s method. The quality of watermarked image in terms of PSNR is much better. The extracted watermarks also have better correlation coefficients.

References

1. Arnolad, M., Schumucker, M., Wolthusen, A.: Techniques and Applications of Digital Watermarking and Content Protection. Artech House, USA (2003)
2. Haj, A.A.: Combined DWT-DCT Digital Image Watermarking. *J. Computer Science* 3, 740–746 (2007)
3. Rao, K., Yip, P.: Discrete Cosine Transform: algorithms, advantages, applications. Academic Press, USA (1990)
4. Potdar, V., Han, S., Chang, E.: A Survey of Digital Image Watermarking Techniques. In: IEEE International Conference on Industrial Informatics, Perth, pp. 709–716 (2005)
5. Wang, R., Lin, C., Lin, J.: Copyright protection of digital images by means of frequency domain watermarking. In: SPIE Conference on Mathematics of Data/Image Coding, Compression and Encryption, USA
6. Do, M.N., Vitterli, M.: M.: The Contourlet Transform: An efficient directional multiresolutional image representation. *IEEE transactions on Image Processing* 14, 2091–2096 (2005)
7. Chandra Mohan, B., Srinivas Kumar, S.: Robust Digital Watermarking Scheme using Contourlet Transform. *J. Computer Science and Network Security* 40(2), 43–51 (2008)
8. Bamerger, R.H., Smith, M.J.T.: A filter bank for the directional decomposition of images: theory and design. *IEEE transactions on Signal Processing* 40, 882–893 (1992)
9. Burt, P.J., Adelson, E.H.: The Laplacian pyramid as a compact image codes. *IEEE transactions on Communications* 31, 532–540 (1983)

10. Jaylakshmi, M., Merchant, S.N., Desai, U.B.: Blind Watermarking in Contourlet Domain with Improved Detection. In: International Conference on Intelligent Information Hiding and Multimedia Signal Processing (2006)
11. Do, M.N., Vitterli, M.: Contoulets: a directional multiresoulational image representation. In: International Conference on Image Processing, pp. 497–501 (2006)
12. Ghannam, S., Abou-Chadi, F.E.Z.: Enhancing Performance of Image Watermarks Using Contorulet Transform. In: National Radio Science Conference, Egypt, pp. 1-9 (2009)
13. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image Quality assessment: From error measurement to structural. *IEEE Transcations on Image Processing* 13(1), 601–612 (2004)
14. Rawat, S., Raman, B.: A New Roubust Watermarking Scheme for Color Images. In: IEEE International Advance Computing Conference, Patiala, pp. 206–209 (2010)
15. Hong, Z.Q.: Algebraic feature extraction of image for recognition. *J. Pattern Recognition* 24, 211–219 (1991)

Face Recognition Using Multi-exemplar Discriminative Power Analysis

Ganesh Bhat¹ and K.K. Achary²

¹ Dept of Electronics, Canara Engineering Collage,
Mangalore-574 219, India
ganeshvbhat@yahoo.com

² Dept of Statistics, Mangalore University,
Mangalore-547199, India
kka@mangaloreuniversity.ac.in

Abstract. Face recognition systems based on holistic approach consider facial images as high-dimensional data points generated from a multi modal distribution. Recent efforts in methods based on this assumption have been in deriving multi-descriptor nonparametric classification systems. However, for problems such as face recognition, identification of multiple descriptors for class representation is not trivial. Most of the recent nonparametric classification techniques solve this problem in a low dimensional feature space using supervised parametric density based clustering methods. In this paper, we investigate two unsupervised non parametric clustering methods namely, Mean Shift technique and Partition Around Medoid clustering, to obtain multiple descriptors in the feature space derived using Discriminant Power Analysis (DPA). Simulation results of proposed nonparametric multi descriptor extension of DPA on the AR and Yale database indicate improvement in recognition rate of DPA over parametric multi descriptor and single descriptor classifiers.

Keywords: Bandwidth selection, discrete cosine transform, density estimation, face recognition.

1 Introduction

Statistical methods used for domain description often assume an underlying statistical law for given patterns. The parameters of the assumed underlying distribution estimated from the dataset are later used to represent the dataset, Mixture of Gaussians (MoG) [1] and Finite mixture model (FMM) [2] are the most commonly used techniques which fall under this category. Another well known method to capture the structure of given dataset is to quantize it in to a set of templates, which can then be used to replace the given data set. Techniques which fall under this category normally use nonparametric density estimators and distance based approaches to cluster the given data in to groups.

Multi-descriptor based nonparametric classification and clustering has received attention because of its practical importance in face recognition problem. Descriptors derived from principal component analysis (PCA), Linear Discriminant Analysis

(LDA), Gabor filters,... etc., have been used with a certain degree of success to face recognition [3]. In most of the recent multi-descriptor techniques, number of descriptors or clusters are determined mainly based on validation for minimal error rate using leave-out validation error criterion.

In recent years discrete cosine transform (DCT) has found wide application as feature extraction tool in face recognition problems [4, 5]. The advantage of using DCT is that the basis functions used are not complex, data-independent and its information packing ability closely approximates the optimal Karhunen-Loeve transform (KLT). DCT hence provides a good compromise between information packing ability and computational complexity. Making use of DCT for feature extraction, recently a method termed as DPA [5] has been proposed and experimentally found to be superior to PCA and LDA in terms of recognition rate and computational requirement.

In this paper, we propose to improve the performance of a computationally simple single descriptor DPA method [4, 5] by extending it to a multi-descriptor based recognition system. Three unsupervised approaches based on a) The Mean estimate of FMM b) Mode selection using nonparametric density approximation c) Medoid selection using partition around medoid (PAM), are investigated, to determine descriptors/exemplars (from here on we term domain descriptors as exemplars) from a particular class in a totally unsupervised manner (with out cross validation involved to determine information on number of clusters/mixtures present).

2 Face Recognition Using DPA

Extracting proper features is crucial for satisfactory design of any pattern classifier system. In pattern recognition literature two types of discrete transforms, statistical and deterministic, have been widely used for feature extraction and data redundancy reduction. The basis vectors of the statistical transforms depend on the statistical specification of the database and different basis vectors are derived for different datasets. Deterministic transforms have invariant basis vectors which are independent of the database. Although statistical transforms have a great ability to remove correlation between data, they have high computational complexity. Also computation of the basis vectors for each given database is needed. Among statistical approaches, PCA and LDA are the two most widely used statistical tools for feature extraction and data representation.

In order to overcome limitations of the PCA and LDA, a variety of modifications have been proposed [6]. The DPA approach [5] is one such approach which selects features (DCT coefficients) with respect to their discrimination power. The DPA utilizes statistical analysis of the given database, assigns each of the features (DCT coefficients) a discrimination parameter and generates a coefficient selection mask to find the best discriminant coefficients for each of the datasets. Dimension reduction in DPA is obtained by retaining only a subset of the calculated coefficients which carry significant amount of discriminant information.

In general, the DCT coefficients of a given image can be divided into two bands, namely low frequencies and high frequencies. Low frequencies are correlated with the illumination conditions and high frequencies represent noise and small variations (details).

It is well known that low frequencies coefficients contain useful information and construct the basic structure of a facial image hence the low frequencies coefficients are more suitable candidates in face recognition [7]. The DCT of an image $f(x,y)$ with dimension $(M \times N)$ is defined by

$$C(u,v) = \frac{2}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \alpha(u)\alpha(v)f(x,y) \cos\left[\frac{(2x+1)u\pi}{2M}\right] \cos\left[\frac{(2y+1)v\pi}{2N}\right] \quad (1)$$

where $0 \leq u \leq M-1, 0 \leq v \leq N-1$ and

$$\alpha(u) = \begin{cases} 1/\sqrt{M} & u=0 \\ \sqrt{2/M} & 1 \leq u \leq M-1 \end{cases}, \quad \alpha(v) = \begin{cases} 1/\sqrt{N} & v=0 \\ \sqrt{2/N} & 1 \leq v \leq N-1 \end{cases}$$

By paying attention to the points stated in the above discussion, the DPA approach selects fixed number of DCT coefficients in the low frequency depending on two attributes: large variation between the classes and small variation within the classes (large discrimination power).

Let the given dataset consist of N classes each class consisting of L image vectors of dimension d , then between class scatter matrix is computed as,

$$S_b = \frac{1}{N} \sum_{j=1}^N (\boldsymbol{\mu}_j - \boldsymbol{\mu})(\boldsymbol{\mu}_j - \boldsymbol{\mu})^T, \quad S_w = \frac{1}{L \times N} \sum_{j=1}^N \sum_{i=1}^L (\mathbf{z}_{ij} - \boldsymbol{\mu}_j)(\mathbf{z}_{ij} - \boldsymbol{\mu}_j)^T \quad (2)$$

$$\text{where, } \boldsymbol{\mu}_j = \frac{1}{L} \sum_{i=1}^L \mathbf{z}_{ij}, \quad \boldsymbol{\mu} = \frac{1}{N} \sum_{j=1}^N \boldsymbol{\mu}_j$$

Where, $\boldsymbol{\mu}$ is called global mean, $\boldsymbol{\mu}_j$ is the mean of the j^{th} class and \mathbf{z}_{ij} represents a feature vectors (lexicographically arranged) of dimension d , obtained from the low frequency DCT coefficients of i^{th} person of j^{th} class. Having represented the principal diagonal elements of S_b and S_w as d dimensional vectors V_b and V_w . Dividing V_b by V_w element wise ($V_b ./ V_w$) gives a measure of discriminating ability of DCT coefficients. Only those DCT coefficients present at location where $V_b ./ V_w$ is high are retained to form the new reduced feature space. We term these coefficients as Most Discriminative Power (MDP) coefficients. After the location of MDP coefficients found for the training set, for a given test image, feature extraction is a simple matter of computing its DCT and retain only the coefficients at the Most Discriminative Information Position (MDIP).

As seen in Fig.1 the cluster obtained for a given class using MDP coefficients exhibit a fair amount of dispersion resulting in marginal overlapping between clusters of different subjects. This motivates us to propose a multi exemplar version of DPA to improve the domain description ability of DPA. However, estimation of multiple exemplars is not trivial, since the learning algorithm needs to estimate a near optimal number of exemplars and then migrate them evenly to the proper regions which often entails them traveling through low likelihood intermediate solutions.



Fig. 1. Clusters of 5 persons formed in a three-dimension feature space constructed using the first three MDP coefficients

We addressed this problem by evaluating two different nonparametric clustering strategies based on 1) representing the modes estimated using non parametric density based approach as exemplars 2) using the medoids derived from a PAM based medoid seeking algorithm as exemplars. We term these approaches as Nonparametric Probability Density Based Approach (NDBA) and Medoid Based Approach (MBA).

3 Non Parametric Unsupervised Learning

3.1 Nonparametric Probability Densities Based Approach (NDBA)

The rationale behind the density estimation based on nonparametric clustering approach is that the feature space can be regarded as empirical probability density function (*pdf*). With the dense regions in the feature space corresponding to local maxima of the *pdf*, estimated as modes of the unknown density. Once the location of a mode is determined the cluster associated with it is delineated based on the local structure of the feature space. Assuming the underlying multivariate distribution of each class as a multimodal distribution, we seek an algorithm to solve the classification/ recognition problem by determining the modes of the distribution to be used as templates/exemplars. Our approach to mode detection is based on the mean shift procedure which has recently become popular due to their simplicity and robustness [8].

The two most popular methods for nonparametric density estimation are the kernel method and the orthogonal series methods. In the kernel method the value of the density $f(x)$ at x is estimated as $\hat{f}(x)$, given by

$$\hat{f}(x) = \frac{1}{\omega n} \sum_{i=1}^n \mathbf{K} \left(\frac{x - x_i}{\omega} \right) \tag{3}$$

Where x_i are samples drawn from a probability distribution with unknown density function f , K is a kernel function, $\hat{\omega}$ is the window-width/ bandwidth.

Similar to the histogram bin-width the bandwidth $\hat{\omega}$ plays an important role in estimation of a multimodal density function, since $\hat{\omega}$ controls the smoothness of the density estimate and the choice of $\hat{\omega}$ is a crucial problem,. A large bandwidth may result in merger of two different modes; where as a small bandwidth will lead to detection of false modes (local maxima). Hence it is necessary to obtain an effective optimal bandwidth for each of the classes from its class distribution.

One can compute the ideal or optimal value of $\hat{\omega}$ by minimizing the mean-square error (MSE) between the true and the estimated densities with respect to $\hat{\omega}$.

$$MSE\{\hat{f}(\mathbf{x})\} = E\left\{ \left[\hat{f}(\mathbf{x}) - f(\mathbf{x}) \right]^2 \right\} \tag{4}$$

The MSE is a function of \mathbf{x} and so the optimal kernel size $\hat{\omega}$ is also a function of \mathbf{x} . In order to minimize the MSE, the best compromise between variance and bias must be selected. It is shown that an optimal constant kernel size independent of \mathbf{x} can be obtained by minimizing either the integral mean-square error or the expected mean-square error. But in practice, one does not have access to the true density function $f(x)$ which is proposed to be estimated. Thus, a number of heuristic approaches can be taken for finding the window width. One such approach is the Least Squares Cross Validation approach using Gaussian kernels [9].The least squares cross-validation considers integrated squared error (ISE) as a distance measure between estimated density \hat{f} and true density f .

$$ISE\{\hat{f}(x)\} = \int \{\hat{f}(x) - f(x)\}^2 dx = \int f^2(x)dx - 2\int \{\hat{f} f\}(x)dx + \int \hat{f}^2(x)dx \tag{5}$$

Cross validation is considered to be a reasonable criterion to choose a bandwidth without having made any assumptions about the unknown density function [9]. A nice feature of the cross-validation method is that the selected bandwidth automatically adapts to the smoothness of $f(\bullet)$. Often one is not only interested in estimating one-dimensional densities, but also multivariate densities. From our previous experience with the one-dimensional case we might consider adapting the kernel density estimator to the d-dimensional case, and write

$$\hat{f}_{\hat{\omega}}(\mathbf{x}) = \frac{1}{\hat{\omega}^d n} \sum_{i=1}^n K^d\left(\frac{\mathbf{x} - \mathbf{x}_i}{\hat{\omega}}\right) = \frac{1}{\hat{\omega}^d n} \sum_{i=1}^n K^d\left(\frac{x_1 - x_{i1}}{\hat{\omega}}, \dots, \frac{x_d - x_{id}}{\hat{\omega}}\right) \tag{6}$$

K^d denoting a multivariate kernel function operating on d arguments. Note that Eq.(6) assumes that the bandwidth $\hat{\omega}$ is the same for each component. If we relax this assumption then we have a vector of bandwidths $\hat{\omega}^d = (\hat{\omega}_1, \dots, \hat{\omega}_d)^T$ and the multivariate kernel density estimator becomes

$$\hat{f}_\omega(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n \frac{1}{\hat{\omega}_1, \dots, \hat{\omega}_d} \mathbf{K}^d \left(\frac{x_1 - x_{i1}}{\hat{\omega}_1}, \dots, \frac{x_d - x_{id}}{\hat{\omega}_d} \right) \tag{7}$$

Hence the multivariate kernel can be written in terms of multiplicative univariate kernel functions

$$\mathbf{K}^d = \mathbf{K}(u_1) \cdot \dots \cdot \mathbf{K}(u_d) \tag{8}$$

The first step in the analysis of a feature space with underlying density is to find the modes of this density; the modes are located among the zeros of the gradient and mean shift procedure is an elegant way to locate these modes. The gradient of density estimator can be written as,

$$\hat{\nabla} f_\omega(x) \equiv \frac{2c_k}{\hat{\omega}^{d+2} n} \sum_{i=1}^n (\mathbf{x} - \mathbf{x}_i) \mathbf{K}' \left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{\hat{\omega}} \right\|^2 \right) \tag{9}$$

denoting $-\mathbf{K}'(\mathbf{x}) = \mathbf{G}(\|\mathbf{x}\|^2)$ yields,

$$\begin{aligned} \hat{\nabla} f_\omega(\mathbf{x}) &= \frac{2c_k}{\hat{\omega}^{d+2} n} \sum_{i=1}^n (\mathbf{x}_i - \mathbf{x}) \mathbf{G} \left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{\hat{\omega}} \right\|^2 \right) = \\ &= \frac{2c_k}{\hat{\omega}^{d+2} n} \left[\sum_{i=1}^n \mathbf{G} \left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{\hat{\omega}} \right\|^2 \right) \right] \left[\frac{\sum_{i=1}^n \mathbf{x}_i \mathbf{G} \left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{\hat{\omega}} \right\|^2 \right)}{\sum_{i=1}^n \mathbf{G} \left(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{\hat{\omega}} \right\|^2 \right)} - \mathbf{x} \right] \end{aligned} \tag{10}$$

The first term in square bracket of Eq.(10) is proportional to the density estimate at \mathbf{x} computed with the kernel \mathbf{G} and the second term is the mean shift, i.e., the difference between weighted mean using the kernel \mathbf{G} for weights and \mathbf{x} . The mean shift vector obtained from the second term always points towards the direction of maximum increase in density .This is more general formulation of the property first remarked by Fukunaga and Hostelter [10, p.535]. In practice, the convergence of mean shift procedure based on kernels such as normal kernel require a large number of steps, therefore one makes use of uniform kernels for which the convergence is finite.

In our experiment the modes of the data points represented in the MDP feature space is obtained using Mean Shift technique with a multivariate kernel of uniform band width. The uniform bandwidth of the multivariate kernel is taken to be the average band width of univariate Gaussian kernels obtained from each of the feature dimension using least square cross validation criterion, so as to minimize the ISE with respect to the smoothness of the estimate, employing mean shift algorithm from arbitrary points simultaneously. The stationary-points/modes to which these iterations converge are later taken as the exemplars of that class for recognition.

3.2 Clustering Using Medoid Based Approach (MBA)

Drawback of distribution and density based techniques for mode selection is that, when large differences in density exist, objects in low density areas will be rejected although they represent legitimate objects. Employing k -medoid-style clustering we aim in finding a set of k representatives among all objects in the data set that best characterize the objects in the data set. Clusters are created by assigning each object to the cluster of the representative (medoid) that is closest to that object. One might wonder why work on developing technique based on k -medoid-style supervised clustering algorithms. The reason is that we believe that medoids are quite useful for class-specific data summarization, because a medoid is the most prototypical object of the members of a cluster. Moreover, algorithms that restrict representatives to objects belonging to the data set, such as k -medoid, explore a smaller solution space if compared with centroid-based clustering algorithms, such as k -means, which searches a much larger set of representatives. Finally, when using k -medoid style clustering algorithms, only an inter-object distance matrix is needed and no “new” distances have to be computed during the clustering process as is the case with k -means.

The clustering procedure PAM [11, chap. 2] takes dissimilarity matrix as input and produces a set of cluster centers or medoids as output. Initially k medoids are randomly chosen out of the set of objects. In the iterative process a non-medoid object is then randomly chosen for replacement with current medoids. Each replacement causes movement of some objects from one cluster to other. Each time a reassignment occurs a difference in square error is contributed to the cost function. If the total cost is negative then replacement of medoid with non-medoid object is good since the actual square error would be reduced. The process is iterated until good replacement of medoid is found. For further details see [11].

In addition to allowing a flexible distance metric, PAM has the advantage of identifying clusters by the medoids. Apart from medoids being robust representations of the cluster centers that are less sensitive to outliers than other cluster profiles, such as the cluster means of k -means PAM tends to separate a large group in to smaller groups to reduce the total sum of distance of all points to the nearest medoid. This robustness is particularly important in the common context that many elements do not belong well to any cluster. We felt that this flexibility is particularly important in applications related to small sample size problem where researchers may be interested in detecting these sample patterns.

One can consider k as given or it can be data-adaptively selected, for example, by using Euclidean and distance maximizing the average silhouette as recommended by Rousseeuw [12]. The silhouette for a given element is calculated as follows. The silhouette of sample vector j is defined by the formula:

$$\begin{aligned}
 \text{Sil}_j(M) &= \frac{b_j - a_j}{\max(a_j, b_j)}, \quad \text{where} \\
 a_j &= \text{avg } d(\mathbf{x}_j, \mathbf{x}_{j'}), \quad j' \in \{i : l_1(\mathbf{x}_i, \mathbf{M}) = (\mathbf{x}_j, \mathbf{M})\}, \\
 b_{jk} &= \text{avg } d(\mathbf{x}_j, \mathbf{x}_{j'}), \quad j' \in \{i : l_1(\mathbf{x}_i, \mathbf{M}) = k\}; \quad b_j = \min_k b_{jk}; \quad k \neq l_1(\mathbf{x}_j, \mathbf{M})
 \end{aligned}
 \tag{11}$$

where K be the number of clusters and let $M=(m_1, \dots, m_k)$ denote any size K collection of the n elements x_i . Given M , we can calculate the dissimilarity $d(x_i, M)$ between each element x_i and each member of M . Note that the largest possible silhouette is 1, which occurs only if there is no dissimilarity within sample vectors j' of a given cluster (i.e.: $a_j = 0$). The other extreme is -1. Heuristically, the silhouette measures how well matched an object is to the other objects in its own cluster versus how well matched it would be if it were moved to another cluster.

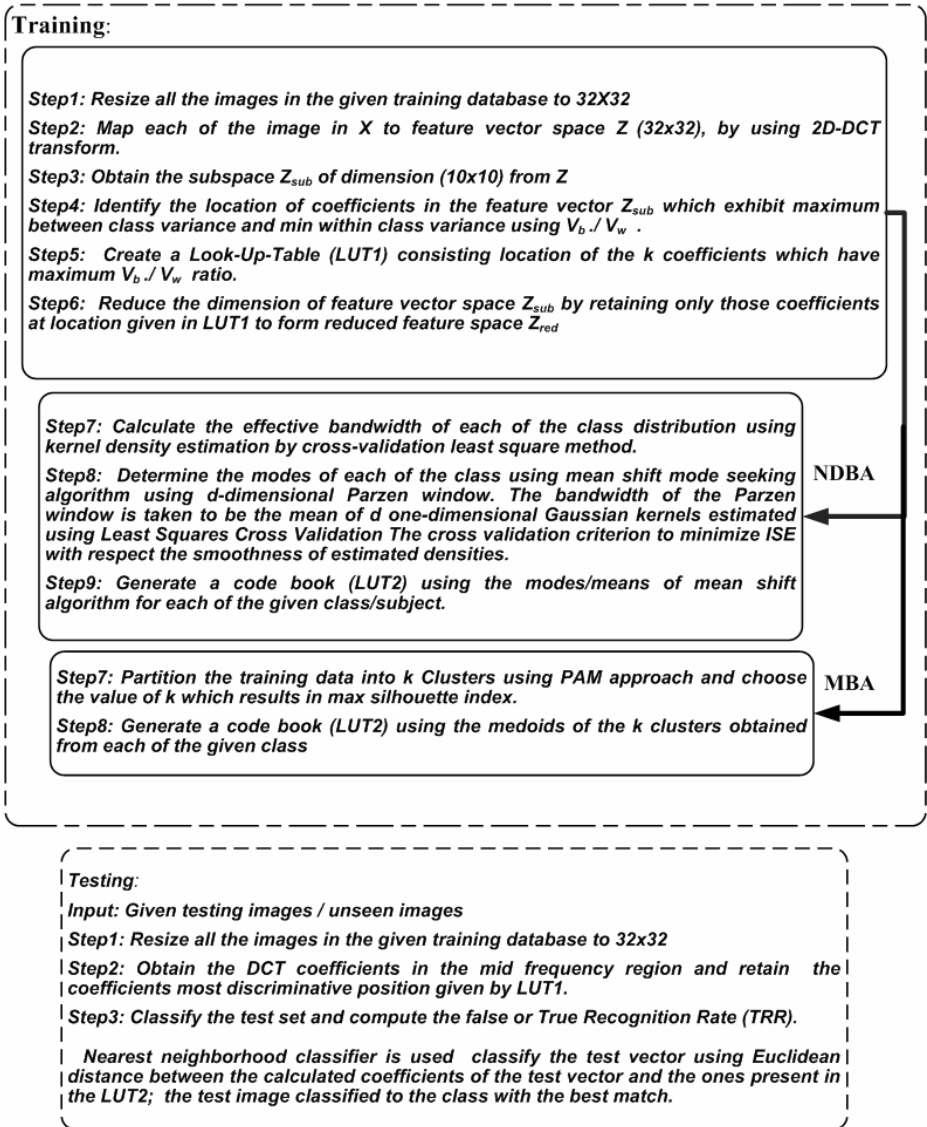


Fig. 2. Flow of the proposed NDBA and MBA techniques

3.3 Experimentation with Unsupervised Learning on Frontal Face Data

In this section we formulate simplified nonparametric multi exemplar face recognition classifier in the MDP feature space using the proposed NDBA and MBA method. Fig.2 gives details on the proposed NDBA and MBA methods. We also give a comparison of these approaches with FMM. The unsupervised FMM proposed by Figueiredo and Jain [2] has been found to be useful for a wide range of unsupervised parametric estimation and mixture estimation problems and distinguishes itself from previous methods [13] by searching for the “best” finite mixture model among the collection of candidate models using minimum message length criterion embedded in to the expectation maximization (EM) algorithm.

3.4 Experimental Procedure

A. In-Database test: To test the learning ability, we use all the images present in the given database as the training set. After learning, we test the algorithm with the same dataset for identification rate.

B. Out-Database test: To test the ability of the algorithm to identify unknown faces with variations/expressions unseen during training, we use the hold out method by excluding all images with identical expression from our training. After learning, we test the performance of the system using holdout images.

3.5 Datasets and Results

In order to evaluate the proposed approaches, our experiments are performed on two benchmark face databases: Yale [14] and AR [15] database. The experiments are programmed in the MATLAB language (version 7). In all of the simulations, the database is initially resized to a size of 32×32 , and is indexed according to each of the expression seen in the database.

A. In-Database test Results

In order to determine the size of low frequency DCT coefficients (dimension of feature vector) which are able to capture the structural information we perform the in-database test using mean of the feature vectors of each class for varying size of low frequency region (i.e. the low frequency window size is gradually increased from 3×3 to 17×17), Fig.3 shows the results of this test for Yale and AR face database.

Based on these results we conclude that a low frequency mask size of 10×10 is sufficient enough to capture the structural information for both the datasets for recognition purpose. In order to determine the number of MDP coefficients required in the low-frequency feature space we perform the in-database test Fig.4 and Fig.5 give the results of this test in terms of true positive recognition rate and average number of exemplars obtained per class using the three multi exemplar approaches along with DPA for AR and Yale face database. Based on recognition rate and average number of exemplars per class (Fig.4B and Fig.5B) it was observed that 7 and 14 features selected at MDIP are sufficient enough to obtain a good recognition rate.

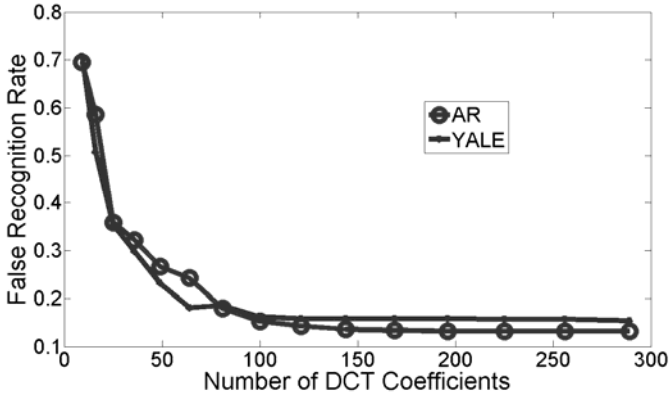


Fig. 3. Plot of false recognition rate obtained using mean feature vector determined by varying the number of coefficients in the low frequency region for Yale and AR database

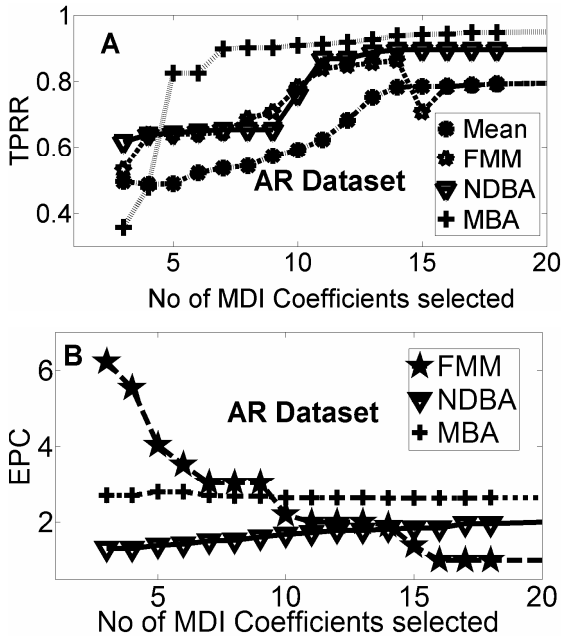


Fig. 4. Plot of number of MDP coefficients Vs (A) True Positive Recognition Rate (TPRR) (B) Average number of exemplars per class (EPC), for AR database using exemplars generated from Mean, FMM, NDBA and MBA

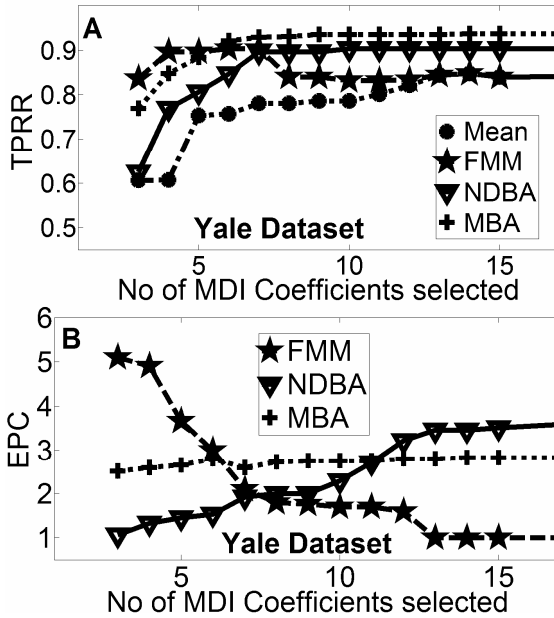


Fig. 5. Plot of number of MDP coefficients Vs (A) True Positive Recognition Rate (TPRR), (B) average number of exemplars per class (EPC), for Yale database using exemplars generated from Mean, FMM, NDBA and MBA

With initial increase in number of selected features the recognition rate of FMM with decrease in average number of exemplars per class however with further increase of selected coefficients beyond an optimal value (7 for Yale and 14 for AR database) there was a decrease in recognition rate with the average number of exemplars converging to one and the FMM approach resulted in the same performance as the mean (DPA) approach.

In case of NDBA and the MBA approach an increase in recognition rate and average number of exemplars is seen with initial increase in number of selected features and saturation effect was observed once the number of selected features increased above the optimal value.

B. Out-Database test Results

In order to test the ability of the algorithm to identify unknown faces with variations/expressions unseen during training a particular expression from all the given subjects is used as the test set and training was conducted with rest of the sample faces. Fig.6 gives the results of this test in terms of recognition rate of a particular expression unseen during training for Yale and AR face datasets.

The expression index 1-11 of Yale face are categorized as Normal, Glasses, Happy, Left Light, Center Light, Surprised, Right Light, Sad, Sleepy, yawn and wink respectively. For AR face datasets expression index 1-13 are categorized respectively as Neutral expression, Smile, Anger, Scream, Right light on, Left light on, All sides

lights on, Wearing sun glasses, Wearing sun glasses and right light on, Wearing sun glasses and left light on, Wearing sun glasses and all light on Wearing scarf, Wearing scarf and left light on, Wearing scarf and right light on.

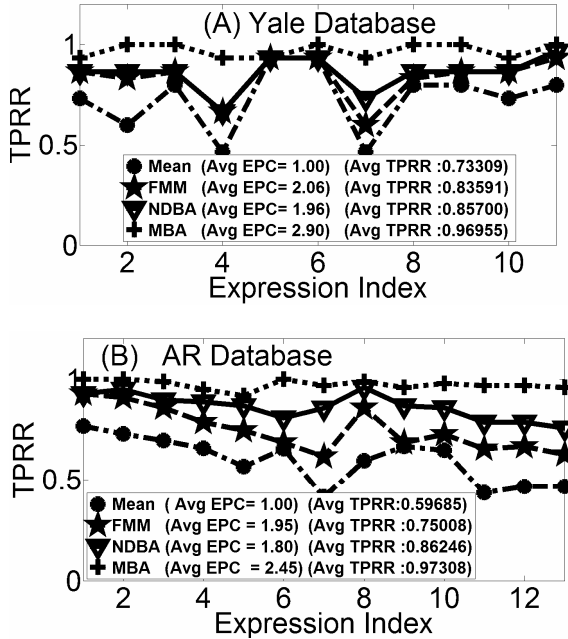


Fig. 6. Plot of True Positive Recognition Rate (TPRR) of a particular expression (Expression Index) of out-database test from Mean, FMMA, NDBA and MBA for (A) YALE database using 7 MDP coefficients (B) AR database using 14 MDP coefficients

4 Conclusion and Future Scope

We have focused on application of Nonparametric unsupervised clustering using a) Exemplar-based clustering using PAM b) Nonparametric density clustering based on mean-shift, to build multi exemplar face recognition system from most discriminative DCT coefficients. The face recognition system developed based on multi exemplar methods displayed a significant improvement in recognition rate over the single exemplar classifier proposed in [10]. By comparing the results of the techniques, conclusion can be drawn that even though the average number of exemplars determined by NDBA and FMM were almost the same, exemplars obtained from NDBA approach serve as better domain descriptors, achieved better performance in terms of recognition rate particularly in case of AR database. It is also seen that selection of number of coefficients in the feature space (dimension of feature space) is more critical in case of FMM for better performance.

The MBA, when compared to NDBA and FMM approach is computationally simpler and displayed superior recognition rate when compared to FMM and NDBA.

The robustness of this approach is particularly because of the ability to more often separately cluster those points that do not belong well to any primary/main clusters. Hence it is found to be better suited for multi-exemplar face recognition task despite the fact that one has to store 25%-35% more number of exemplars per class in MBA when compared to FMM or NDBA. Exemplars determined by unsupervised clustering have displayed better results over the single exemplar technique in the MDP feature space. Further work is needed to verify this result in feature space derived using other methods.

References

1. Hastie, T., Tibshirani, R.: Discriminant analysis by Gaussian mixtures. *Journal of the Royal Statistical Society* 58, 155–176 (1996)
2. Figueiredo, M., Jain, A.K.: Unsupervised learning of finite mixture models. *IEEE Transaction on Pattern Analysis and Machine Intelligence* 24(3), 381–396 (2002)
3. Kevin, S., Chellappa, R.: Multiple-exemplar discriminant analysis for face recognition. In: *Proc. of the 17th International Conference on Pattern Recognition*, Cambridge, UK, pp. 191–194 (2004)
4. Bhat, G., Achary, K.K.: Finding prototype vectors in high dimensional data sets with application to face recognition. In: *Proceedings of Second international Conference on Cognition and Recognition*, Mysore, India, pp. 179–186 (2008)
5. Saeed, D., Masoumeh, P.G., Ali, A.: A Feature extraction using discrete cosine transform and discrimination power analysis with a face recognition technology. *Pattern Recognition* 43, 1431–1440 (2010)
6. Zuo, W., Zhang, D., Yang, J., Wang, K.: BDPCA plus LDA: A novel fast feature extraction technique for face recognition. *IEEE Transactions on Systems, Man, and Cybernetics—Part b: Cybernetics* 36(4) (2006)
7. Dabbaghchian, S., Aghagolzadeh, A., Moin, M.S.: Reducing the effects of small sample size in DCT domain for face recognition. In: *International Symposium on Telecommunication IST*, Tehran, August 27–28, pp. 634–638 (2008)
8. Hager, G., Dewan, M., Stewart, C.: Multiple kernel tracking with SSD. In: *Proc. IEEE Conf. Comp. Vision Pattern Recognition*, Washington, D.C, vol. 1, pp. 790–797 (2004)
9. Hall, P., Marron, J.S., Park, B.U.: Smoothed cross-validation. *Probability Theory and Related Fields* 92, 1–20 (1992)
10. Fukunaga, K., Hostetler, L.D.: The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition. *IEEE Trans. Info. Theory* IT-21, 32–40 (1975)
11. Kaufman, L., Rousseeuw, P.J.: *Finding Groups in Data: an Introduction to Cluster Analysis*. John Wiley & Sons, Chichester (1990)
12. Rousseeuw, P.J.: Silhouette: a graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* 20, 53–65 (1987)
13. McLachlan, G.J., Peel, D.: *Finite Mixture Models*. Wiley, New York (2000)
14. Yale University,
<http://cvc.yale.edu/projects/yalefaces/yalefaces.html>
15. Martinez, A.M., Benavente, R.: The AR face database. *CVC Tech. Report # 24* (1998)

Efficient Substitution-Diffusion Based Image Cipher Using Modified Chaotic Map

I. Shatheesh Sam¹, P. Devaraj², and R.S. Bhuvaneshwaran¹

¹ Ramanujan Computing Centre, College of Engineering, Guindy
Anna University, Chennai, India

² Department of Mathematics, College of Engineering, Guindy,
Anna University, Chennai, India
shatheeshsam@yahoo.com

Abstract. An efficient substitution-diffusion based image cipher using modified chaotic map is proposed using chaotic maps. The proposed scheme consists of permutation which uses the odd key values and XORing with first chaotic key for confusion. Nonlinear diffusion is obtained by circular shift and XORing with second chaotic key. It is used to improve the security against the known/chosen-plaintext attack. Finally, confusion and diffusion are obtained using the alternative zig-zag diffusion of adjacent pixels and XORing with the third chaotic key. The number of rounds in the steps are controlled by combination of pseudo random sequence and original image. The security and performance of the proposed image encryption technique have been analyzed thoroughly using statistical analysis, differential analysis, key space analysis and entropy analysis. The experimental results illustrate that the performance is more secure and fast.

Keywords: Permutation, Alternative Zig-Zag, Modified Chaotic Map.

1 Introduction

With advancements in digital communication technology and the growth of computer power and storage, the difficulties in ensuring individuals privacy become increasingly challenging. In any communication system, including satellite and internet, it is almost impossible to prevent unauthorized people from eavesdropping. When information is broadcasted from a satellite or transmitted through the internet, there is a risk of information interception. Security of image and video data has become increasingly important for many applications including video conferencing, secure facsimile, medical and military applications. The conventional encryption techniques [2] are not reliable for image encryption due to high redundancy, strong correlation and high computation complexity. The chaos based cryptosystems[1] are suitable for the secure transmission of images due to the intrinsic features of chaotic systems such as ergodicity, mixing property, sensitivity to initial conditions and system parameters. It can be considered analogous to ideal cryptographic properties such as confusion and diffusion properties. Hence, many chaos [4-5] based encryption systems have been proposed in

the last two decades. The one dimensional [3] chaos system has the advantages of simplicity and high security. Many studies were proposed to adapt and improve it. Some of the cryptanalysis techniques [7-8] are suggested to break the scheme and reduce the flaws in the algorithm design.

In [6], a new image encryption scheme based on the logistic and standard maps was proposed, where the two maps are used to generate a pseudo random number sequence (PRNS) controlling two kinds of encryption operations. Rhouma et al. [9] reported that the above scheme is not secure since an equivalent key can be obtained from known/chosen plaintext and the corresponding ciphertext.

In this paper, an efficient substitution-diffusion based image cipher using modified chaotic map is suggested to improve the security level. The algorithm uses significant features such as sensitivity to initial condition, permutation of keys, modified chaotic maps, alternative zig-zag diffusion. The nonlinearity is used to overcome the main limitation of the Patidar et al. The rest of this paper is organized as follows. Section 2 introduces logistic map. In section 3, the image encryption based on modified chaotic map is proposed including a new algorithm. In section 4, the security of new algorithm is analyzed. Finally, the conclusions are discussed in section 5.

2 Logistic Map

Logistic map is the most widely used classical map. It is very simple and deterministic, but it has complicated dynamic behavior. The logistic map is defined as follows:

$$x_{n+1} = \mu x_n(1 - x_n)$$

where μ is system parameter, $0 < \mu \leq 4$ and x_n is a floating number in $(0,1)$, $n = 0, 1, 2, 3 \dots$. When $\mu > 3.587696363$, this system becomes chaotic in behavior. Its bifurcation diagram is shown in Fig. 1.

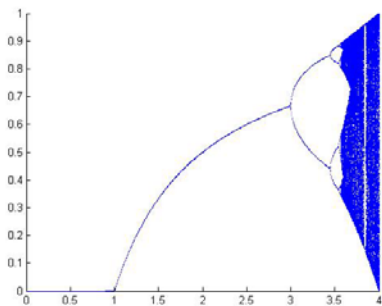


Fig. 1. Bifurcation Diagram of Logistic Map

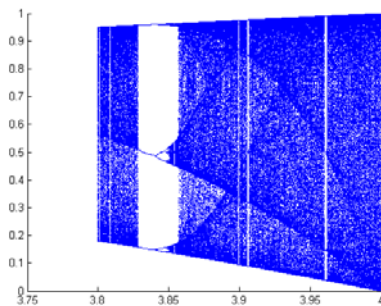


Fig. 2. Blank Window for the Logistic Map

Among the special features of logistic map, its high sensitivity to initial value and parameter makes it suitable for image encryption. Though, the logistic map is better for image encryption which has some common problems such as stable windows, blank windows, uneven distribution of sequences and weak key [10]. The blank window is more serious problem than others. Fig. 2 illustrates that the blank window appears when $\mu = 3.828$.

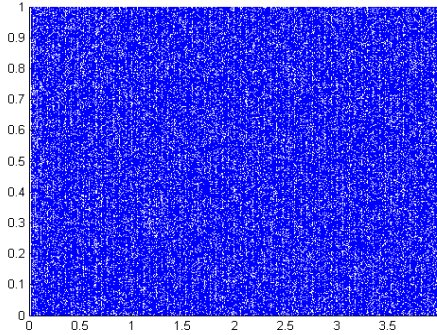


Fig. 3. Distribution of Sequence for the Modified Chaotic Map

New types of modified chaotic logistic maps have been proposed in this paper to alleviate the problems in the logistic map. The maps are mixed together so as to achieve larger key space and to attain chaotic behavior. We have attempted to improve it by chaotic transformation. The proposed modified chaotic logistic maps are defined and keys are generated in the next section. Thus, the proposed modified chaotic logistic map does not have security issues which are present in the logistic map. Moreover, the resulting chaotic sequences are uniformly distributed (see the Fig. 3) and the key size has been increased greatly.

3 Proposed Scheme

The plain image is stored in a two dimensional array of $M \times N$ pixels. $Plainimage = \{Gray_{i,j}\}$, where $1 \leq i \leq H$ and $1 \leq j \leq W$, H and W are height and width of the original image in pixels

3.1 Key Generation

The modified chaotic map and the keys have been generated in the following way:

$$\begin{aligned}
 &for\ i = 1\ to\ H \\
 &for\ j = 1\ to\ W \\
 &x_{i,j+1} = (3.833417 \times k_1 \times (1 + x_{i,j}) + y_{i,j}) \bmod 1 \\
 &y_{i,j+1} = (3.999283 \times k_2 \times y_{i,j} \times \sin(y_{i,j}) \times (1/(1 - (x_{i,j+1})^2))) \bmod 1 \\
 &z_{i,j+1} = (3.669943 \times k_3 \times x_{i,j+1} \times y_{i,j+1} \times z_{i,j}) \bmod 1
 \end{aligned}$$

```

     $KX_{i,j} = \lfloor x_{i,j+1} \times 256 \rfloor$ 
     $KY_{i,j} = \lfloor y_{i,j+1} \times 256 \rfloor$ 
     $KZ_{i,j} = \lfloor z_{i,j+1} \times 256 \rfloor$ 
  end
   $x_{i+1,1} = x_{i,j+1}$ 
   $y_{i+1,1} = y_{i,j+1}$ 
   $z_{i+1,1} = z_{i,j+1}$ 
  end

```

where $|k_1| > 23.7$, $|k_2| > 27.3$, $|k_3| > 25.9$ respectively. To increase the key size we can use k_1, k_2, k_3 as another set of keys. Along with the key k_i is the distribution of the sequences becomes better. $KX_{i,j}, KY_{i,j}, KZ_{i,j}$ are the set of chaotic keys.

3.2 Initial Permutation

The chaos based image encryption schemes are mainly consisting of image pixel permutation stage otherwise called as confusion stage and pixel value diffusion stage. Generally, the confusion effect is considered by permutation step. The method is defined by:

$$C[i, j] = P[1 + (23 \times i + 3) \bmod 256, 1 + (51 \times j + 3) \bmod 256] \oplus KX_{i,j}$$

where $P[i, j]$ represents the $(i, j)^{th}$ pixel of the original image and $C[i, j]$ denotes the $(i, j)^{th}$ pixel of the cipher image. $KX_{i,j}$ is the first chaotic key and XORing with permuted values. This method also uses two constant values to improve the pixel scrambling of image.

3.3 Nonlinear Diffusion

Diffusion refers to the property that redundancy in the statistics of the plaintext is dissipated in the statistics of the cipher text. The grey diffusion is done by 3 bit circular shift method. The resultant values are XORing with second chaotic key. The combination of 3 bit circular shift and XORing makes the encryption operation nonlinear and hence the system becomes strong against known/chosen plaintext attack. The procedure for the nonlinear diffusion is as follows:

```

  for  $i = 1$  to  $H$ 
    for  $j = 1$  to  $W$ 
       $C_{i,j} = (P_{i,j} \ggg 3) \bmod 256$ 
       $C_{i,j} = C_{i,j} \oplus KY_{i,j}$ 
    end
  end

```

where $KY_{i,j}$ is the second chaotic key.

3.4 Alternative Zig-Zag Diffusion

In this, we read the values in the alternative zig-zag (see Fig. 4) manner as follows:

3.5.1 Inverse Nonlinear Diffusion

The inverse nonlinear diffusion is done by 3 bit reverse circular shift method. The resultant values were XORing with same second chaotic key. The procedure for the inverse nonlinear diffusion is as follows:

```

for i = 1 to H
  for j = 1 to W
     $C_{i,j} = (C_{i,j} \lll 3) \bmod 256$ 
     $P_{i,j} = C_{i,j} \oplus KY_{i,j}$ 
  end
end

```

3.5.2 Inverse Permutation

The permutation is replaced by inverse permutation. The inverse method is described by:

$$P[i, j] = C[1 + (167 \times i - 4) \bmod 256, 1 + (251 \times j - 4) \bmod 256] \oplus KX_{i,j}$$

The original image can be recovered once the above decryption process is completed.

4 Security and Performance Analysis

A good encryption scheme should be robust against all kinds of cryptanalytic, statistical and brute force attacks. Some experimental results are given in this section to demonstrate the efficiency of our scheme. All the experiments are performed on a PC with Intel Core 3.0GHz CPU, 4GB RAM with Windows Vista Business Edition. The compiling environment is MATLAB 7.4.

4.1 Statistical Analysis

In order to resist the statistical attacks, which are quite common now-a-days, the encrypted images should possess certain random properties. A detailed study has been undertaken and the results are summarized. Different images have been tested, and we found that the results are similar.

4.2 Histogram Analysis

Histograms may reflect the distribution information of the pixel values of an image. An attacker can analyze the histograms of an encrypted image by using some attacking algorithms to get some useful information of the original image.

Thus, the histograms of an encrypted image should be as smooth and evenly distributed as possible, and should be very different from that of the plaintexts. Fig. 5 shows a comparison of the histograms between plaintext and encrypted images.

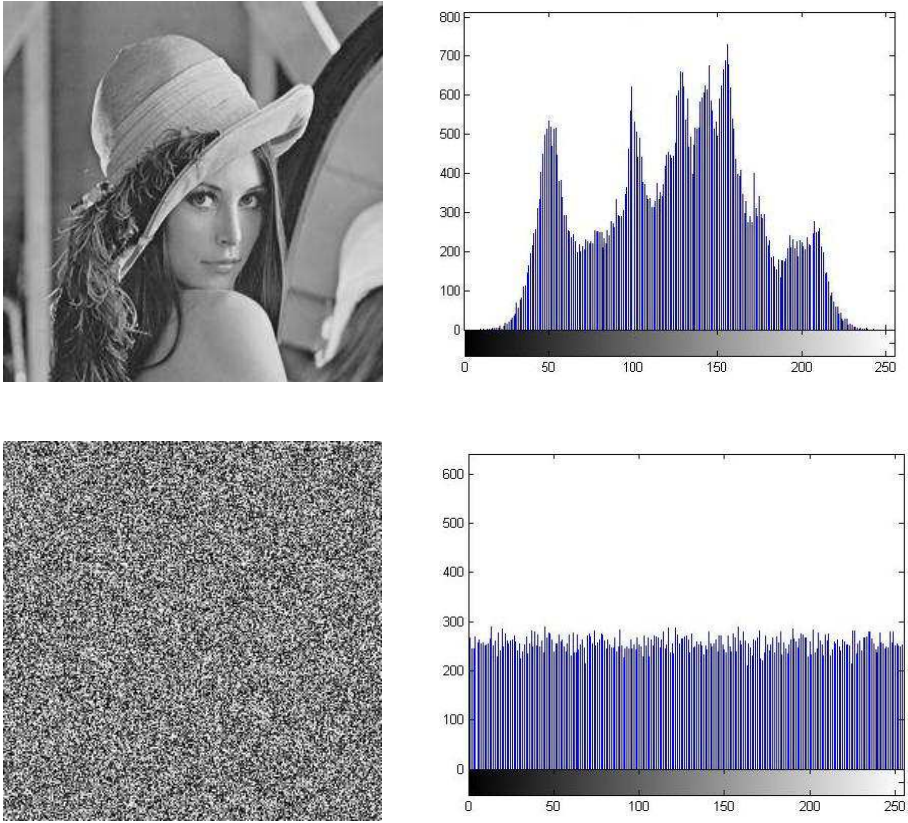


Fig. 5. Histogram Analysis of Plain Image and Cipher Image

4.3 Correlation Coefficient Analysis

We use the 256 grey levels image Lena (256× 256 pixels) as the original image. Experiment shows that image scrambling effect is inverse to the correlation coefficient function of the adjacent pixels. Correlation coefficient function is used as follows.

$$D(x) = \frac{1}{k} \sum_{i=1}^k [x_i - E(x)]^2$$

where x is the grey value of pixel point; k is the number of pixel point; $E(x)$ is mathematical expectation of x and $D(x)$ is variance of x .

$$Conv(x, y) = \frac{1}{k} \sum_{i=1}^k [x_i - E(x)][y_i - E(y)]$$

where x is grey value of the former pixel point; y is grey value of the latter pixel point;

$cov(x, y)$ is the covariance of x, y .

$$r_{xy} = \frac{cov(x, y)}{\sqrt{D(x)}\sqrt{D(y)}}$$

where r_{xy} is the related coefficients. Carrying out the experimental analysis on the adjacent pixel points of the primitive image, the final result is presented in Table 1.

Table 1. Correlation Coefficients of Two Adjacent Pixels

Direction	Plain image	Cipher image
Horizontal	0.9386	0.0015
Vertical	0.9689	0.0042
Diagonal	0.9138	0.0031

4.4 Differential Analysis

Differential attack would become ineffective even if a single pixel change in the plaintext, causes a significant difference in the ciphertext. In order to measure this capability quantitatively, the following measures are usually used: number of pixels change rate (NPCR) and unified average changing intensity (UACI). They are defined as follows:

$$D_{ij} = \begin{cases} 1, & \text{if } C_{ij} \neq C'_{ij} \\ 0, & \text{otherwise} \end{cases}$$

The NPCR is defined as

$$NPCR = \frac{\sum_{i,j} D_{ij}}{M \times N} \times 100\%$$

The UACI is defined as

$$UACI = \frac{1}{M \times N} \left[\sum_{i,j} \frac{C_{ij} - C'_{ij}}{255} \right] \times 100\%$$

where C_{ij} and C'_{ij} are the two ciphertexts at position (i, j) whose corresponding plaintexts have only one pixel difference and M and N are the number of rows and columns of images. The results of NPCR and UACI are listed in Table 2.

In order to assess the influence of changing a single pixel in the original image on the encrypted image, the $NPCR$ and the $UACR$ are computed in the proposed scheme. The results show that a small change in the original image will result in a significant difference in the cipherimage. So the scheme proposed has a high capability to resist anti differential attack.

Table 2. Sensitivity to Ciphertext

	NPCR%	UACI%
Lena	99.6136	33.4897
Baboon	99.6241	33.4813
House	99.6298	33.4878
Tree	99.6306	33.4829

4.5 Key Space Analysis

There are series of improved chaotic maps parameters, initial values, and k_i values that can also be used as keys in our scheme. The key space is in the range between 192 to 420 bits.

4.6 Information Entropy Analysis

Information entropy is one of the criteria to measure the strength of the cryptosystem in symmetric cryptosystem. The entropy $H(m)$ of a message m can be calculated as

$$H(m) = \sum_{i=0}^{2^N-1} p(m_i) \log \frac{1}{p(m_i)}$$

where $p(m_i)$ represents the probability of occurrence of symbol m_i and \log denotes the base 2 logarithm. If there are 256 possible outcomes of the message m with equal probability, it is considered as random. In this case, $H(m) = 8$, is an ideal value. In the final round of proposed scheme, it is found that the value is 7.9954. This means that information leakage in the encryption process is negligible and the encryption system is secure upon entropy attack.

5 Conclusion

In this paper, an efficient substitution-diffusion based image cipher using modified chaotic map has been proposed. The proposed cipher provides good confusion and diffusion properties that ensures extremely high security. Confusion and diffusion have been achieved using permutation, nonlinear diffusion and alternative zig-zag diffusion. This scheme is immune to various types of cryptographic attacks like known/chosen plain text and brute force attacks. We have carried out statistical analysis, differential analysis key space analysis and entropy analysis to demonstrate the security of the new image encryption procedure. Based on the various analyzes, it has been shown that the proposed scheme is secure and fast.

Acknowledgment. The first author is partly supported by the All India Council for Technical Education, New Delhi, India.

References

1. Matthew, R.: On the derivation of a chaotic encryption algorithm. *Cryptologia* 8, 29–42 (1989)
2. Schneier, B.: *Applied cryptography: protocols algorithms and source code in C*. Wiley, New York (1996)
3. Baptista, M.S.: Cryptography with chaos. *Phys. Lett. A* 240, 50–54 (1998)
4. Chen, G.R., Mao, Y.B., Charles, K.C.: A symmetric image encryption scheme based on 3D chaotic cat maps. *Chaos, Solitons & Fractals* 21, 749–761 (2004)
5. Wang, Y., Wong, K.-W., Liao, X., Xiang, T., Chen, G.: A chaos-based image encryption algorithm with variable control parameters. *Chaos. Solitons Fract.* 83, 1773–1783 (2009)
6. Patidar, V., Pareek, N.K., Sud, K.K.: A new substitution-diffusion based image cipher using chaotic standard and logistic maps. *Commun. Nonlinear Sci. Numer. Simulat.* 14, 3056–3075 (2009)
7. Alvarez, G., Shujun, L.: Cryptanalyzing a nonlinear chaotic algorithm (NCA) for image encryption. *Commun. Nonlinear Sci. Numer. Simulat.* 14, 3743–3749 (2009)
8. Chengqing, L., Shujun, L., Asim, M., Alvarez, G., Chen, G.: On the security defects of an image encryption scheme. *Image Vis Comput.* 27, 1371–1381 (2009)
9. Rhouma, R., Solak, E., Belghith, S.: Cryptanalysis of a new substitution-diffusion based image cipher. *Commun. Nonlinear Sci. Numer. Simulat.* 15, 1887–1892 (2010)
10. Jianquan, X., Chunhua, Y., Qing, X., Lijun, T.: An Encryption Algorithm Based on Transformed Logistic Map. In: *IEEE International Conference on Network Security, Wireless Communications and Truusted Computing*, pp. 111–114 (2009)

Non Local Means Image Denoising for Color Images Using PCA

P.A. Shyjila and M. Wilsy

Department of Computer Science, University of Kerala, Thiruvananthapuram
pashijila@gmail.com, wilsyphilipose@hotmail.com

Abstract. The goal of image denoising is to remove unwanted noise from an image. There are various methods for image denoising. The proposed algorithm is a variation of the nonlocal means (NLM) image denoising algorithm that uses principal component analysis (PCA) to achieve a higher accuracy while reducing computational load. Image neighborhood vectors are first projected onto a lower dimensional subspace using PCA. For color images RGB image neighborhood vectors are formed by concatenating image neighborhoods in the three color channels into a single vector. The dimensionality of this subspace is chosen automatically using parallel analysis. Consequently, neighborhood similarity weights for denoising are computed using distances in this subspace rather than the full space. The accuracy of NLM and the proposed algorithm are examined with respect to the choice of image neighborhood and search window sizes. Finally, we present a quantitative and qualitative comparison of the proposed algorithm versus NLM image denoising algorithm.

Keywords: Image denoising, nonlocal means (NLM), parallel analysis, principal component analysis, principal neighborhood.

1 Introduction

The goal of image denoising methods is to recover the original image from a noisy image. The best simple way to model the effect of noise on a digital image is to add a gaussian white noise. Several methods have been proposed to remove the noise and recover the true image. In most of them, denoising is achieved by averaging. This averaging may be performed locally (eg: the Gaussian smoothing model, the anisotropic filtering and the neighborhood filtering), by the calculus of variations (eg: the Total Variation minimization), or in the frequency domain (eg: the empirical Wiener filters and wavelet thresholding methods).

The denoising methods should not alter the original image. Now, most denoising methods degrade or remove the fine details and texture of the original method. Many models used in denoising applications have been based on the assumption of piecewise smoothness [2],[3] and [4]. This type of model is too simple to capture the textures present in a large percentage of real images. This drawback has limited the performance of such models, and motivated data driven representations. One data-driven strategy is to use image neighborhoods or patches as a feature vector for representing local structure. Image neighborhoods are rich enough to capture the local

structures of real images. This representation has been used as a basis for image denoising, for texture synthesis, and for texture segmentation. The image neighborhood feature vector is typically high dimensional. For instance, it is 49 dimensional if 7×7 neighborhoods are used. Hence, the computation of similarities between feature vectors incurs a large computational cost. By projecting image neighborhood vectors into a lower dimensional subspace using Principal Component Analysis (PCA) we can reduce the computational complexity of methods that rely on image neighborhood information.

Principal Neighborhood Dictionaries for Non Local Means image denoising algorithm (PND) [1] is a variation of the nonlocal means (NLM) image denoising algorithm that uses principal component analysis (PCA) to achieve a higher accuracy while reducing computational load. We extend this method by applying it for color images. The nonlocal means (NLM) image denoising algorithm averages pixel intensities using a weighting scheme based on the similarity of image neighborhoods [6]. The nonlocal means (NLM) image denoising algorithm using PCA is a very similar approach in which image neighborhood vectors are projected to a lower dimensional subspace using principal component analysis (PCA). For color images, RGB image neighborhood vectors are formed by concatenating image neighborhoods in the three color channels into a single vector. Then, the neighborhood similarity weights for denoising are computed from distances in this subspace resulting in significant computational savings. More importantly, this approach results in increased accuracy over using the full-dimensional ambient space [9], [10].

One disadvantage of the approach in [10] is the introduction of a new free parameter to the algorithm—the dimensionality of the PCA subspace. This approach proposes an automatic dimensionality selection criteria using parallel analysis [15] that eliminates this free parameter. The parallel analysis method compares the eigen values of the data covariance matrix to eigen values of the covariance matrix of an artificial data set. The method for creating artificial data set is described in section 3. The detailed discussion of the related works are provided in the next section. We provide a detailed discussion of the proposed method in section 3. Finally in section 4 we provide the results and analysis. In this section we compare the proposed method and the NLM algorithm for color images.

2 Related Work

Buades *et al.* introduced the NLM image denoising algorithm which averages pixel intensities weighted by the similarity of image neighborhoods [6]. Image neighborhoods are typically defined as 5×5 , 7×7 , or 9×9 square patches of pixels which can be seen as 25, 49 or 81 dimensional feature vectors, respectively. Then, the similarity of any two image neighborhoods is computed using an isotropic Gaussian kernel in this high-dimensional space. Finally, intensities of pixels in a search-window centered around each pixel in the image are averaged using these neighborhood similarities as the weighting function.

Mahmoudi and Sapiro have proposed a method to improve the computational efficiency of the NLM algorithm [28]. Their patch selection method removes unrelated neighborhoods from the search-window using responses to a small set of predetermined

filters such as local averages of gray value and gradients. Unlike [28] the lower dimensional vectors computed in [9], [10] and [11] are data-driven. Additionally, in [9] and [10], the lower dimensional vectors are used for distance computation rather than patch selection.

In the above methods, the computational complexity is very high because of high dimensionality image neighborhoods. In Principal Neighborhood Dictionaries for Non Local Means Image Denoising Algorithm (PND) [1], this computational complexity is reduced by projecting image neighborhood vectors into a lower dimensional subspace using Principal Component Analysis (PCA). We extend this method by applying it for color images.

Principal component analysis of neighborhoods have previously been used for various image processing tasks. PCA of image neighborhoods was used for denoising [5]. However, in that work, PCA is computed for local collections of image neighborhood samples and denoising is achieved by direct modification of the projection coefficients.

In this algorithm, PCA is computed once, globally rather than locally. This results in a computationally more efficient algorithm. Furthermore, a nonlocal means averaging scheme is used rather than direct modification of projection coefficients.

There are various methods for determining the number of components to retain in data analysis. Parallel Analysis, originally proposed by Horn [15], is one of the most successful methods for determining the number of true principal components. Improvements to the original parallel analysis method have also been proposed. In this algorithm, the method used in [1] is used for subspace dimensionality selection.

3 Proposed Method

The proposed method is a variation of the nonlocal means (NLM) image denoising algorithm. The nonlocal means (NLM) image denoising algorithm averages pixel intensities using a weighting scheme based on the similarity of image neighborhoods. The computational complexity of NLM is very high because of high dimensionality image neighborhoods. The proposed method and PND [1] are very similar approaches in which image neighborhood vectors are projected to a lower dimensional subspace using principal component analysis (PCA). For color images, RGB image neighborhood vectors are formed by concatenating image neighborhoods in the three color channels into a single vector. Then, the neighborhood similarity weights for denoising are computed from distances in this subspace resulting in significant computational savings. More importantly, this approach results in increased accuracy over using the full-dimensional ambient space [9], [10].

The nonlocal means (NLM) image denoising algorithm is discussed in section 3.1. In section 3.2, the proposed method is given. The method for determining the subspace dimensionality is given in section 3.3. Finally, section 3.4 describes how we select the size of search window.

3.1 Non Local Means Algorithm

Starting from a discrete image u , a noisy observation of u at pixel i is defined as $v(i) = u(i) + n(i)$. Let N_i denote a $r \times r$ square neighborhood centered around pixel

i. Also, let $y(i)$ denote the vector whose elements are the gray level values of v at pixels in N_i . Finally, S_i is a square search-window centered around pixel i . Then, the NLM algorithm [6] defines an estimator for $u(i)$ as

$$NL(i) = \sum_{j \in S_i} \frac{1}{Z(i)} e^{-\|y(i)-y(j)\|^2/h^2} v(j) \tag{1}$$

where $Z(i) = \sum_{j \in S_i} e^{-\frac{\|y(i)-y(j)\|^2}{h^2}}$ is a normalizing term. The smoothing kernel width parameter h controls the extent of averaging. For true nonlocal means, the search window S_i needs to be the entire image for all i , which would give rise to global weighted averaging. However, for computational feasibility, S_i has traditionally been limited to a square window of modest size centered around pixel i . This is the limited-range implementation of the NLM algorithm as proposed in the pioneering work by [6]. For instance, a 21×21 window is used in [6] whereas a 7×7 window is used in [9].

3.2 Nonlocal Means Image Denoising for Color Images Using PCA

For color images, $y(i)$'s (image neighborhood vectors) are formed by concatenating image neighborhoods in the three color channels into a single vector. Also in this approach, the distances $\|y(i) - y(j)\|^2$ in equation (1) are replaced by distances computed from projections of y onto a lower dimensional subspace determined by PCA. Let Ω denote the entire set of pixels in the image. Also, let Ψ be a randomly chosen subset of Ω . Treating $y(i)$ as observations drawn from a multivariate random process, we can estimate their covariance matrix as

$$C_y = \frac{1}{\Psi} (y(i) - \bar{y})(y(i) - \bar{y})^T \tag{2}$$

where $\bar{y} = \frac{1}{|\Psi|} \sum_{i \in \Psi} y(i)$ is the sample mean and $|\Psi|$ is the number of elements in the set Ψ . A small subset $\Psi \subseteq \Omega$ is typically sufficient to accurately estimate the covariance matrix and results in computational savings. The dimensionality of a $r \times r$ neighborhood vector is r^2 . For simplicity of notation, let $M = r^2$. Then C_y is a $M \times M$ matrix. Let $\{b_p : p = 1 : M\}$ be the eigenvectors of C_y , i.e., the principal neighborhoods, sorted in order of descending eigen values. Let the d -dimensional PCA subspace be the space spanned by $\{b_p : p = 1 : d\}$. Then the projections of the image neighborhood vectors onto this subspace is given by

$$y_d(i) = \sum_{p=1}^d \langle y(i), b_p \rangle b_p \tag{3}$$

where $\langle y(i), b_p \rangle$ denotes the inner product of the two vectors.

Let $f_d(i) = [\langle y(i), b_1 \rangle \dots \langle y(i), b_d \rangle]^T$ be the d-dimensional vector of projection coefficients. Then, due to the orthonormality of the basis functions

$$\|y_d(i) - y_d(j)\|^2 = \|f_d(i) - f_d(j)\|^2 \tag{4}$$

Finally, define a new family of estimators for $d \in [1, M]$

$$\hat{u}_d(i,1) = \sum_{j \in S_i} \frac{1}{Z_d(i)} e^{-\frac{\|f_d(i) - f_d(j)\|^2}{h^2}} v(j,1) \tag{5}$$

$$\hat{u}_d(i,2) = \sum_{j \in S_i} \frac{1}{Z_d(i)} e^{-\frac{\|f_d(i) - f_d(j)\|^2}{h^2}} v(j,2) \tag{6}$$

$$\hat{u}_d(i,3) = \sum_{j \in S_i} \frac{1}{Z_d(i)} e^{-\frac{\|f_d(i) - f_d(j)\|^2}{h^2}} v(j,3) \tag{7}$$

where $Z_d(i) = \sum_{j \in S_i} e^{-\frac{\|f_d(i) - f_d(j)\|^2}{h^2}}$ is the new normalizing term.

3.3 Automatic Subspace Dimensionality Selection

The original parallel analysis method [15] compares the eigen values of the data covariance matrix to eigen values of the covariance matrix of an artificial data set. This artificial data set is generated by drawing samples from a multivariate normal distribution with the same dimensionality M , the same number of observations $|\Psi|$, and the same marginal standard deviations as the actual data. Let λ_p for $1 \leq p \leq M$ denote the eigen values of C_y sorted in descending order. Similarly, let α_p denote the sorted eigen values of the artificial data covariance matrix. Parallel analysis estimates data dimensionality as

$$d = \max\{1 \leq p \leq M \mid \lambda_p \geq \alpha_p\} \tag{8}$$

The intuition is that the P_3 is a threshold for λ_p below which the p^{th} component is judged to have occurred due to chance.

An improvement to parallel analysis is to use Monte Carlo simulations to generate the artificial data which removes the assumption of normal distribution. This algorithm generates the artificial data by randomly permuting each element of the neighborhood vector across the sample Ψ . Let $y_{i,k}$ denote the k^{th} element of the neighborhood vector $y(i)$. For each k generate a random permutation $j(i)$ of the sequence $i = 1 : |\Psi|$ and let

$w_{i,k} = y_{j(i),k}$. Then, the random vectors $w(i)$ are composed from the elements $w_{i,k}$. The artificial eigen values α are computed from the covariance matrix of w . This method for computing the artificial covariance matrix keeps the marginal distributions intact while breaking any interdependencies between them.

Several researchers have previously discussed that parallel analysis has a strong tendency to underestimate the number of components in data where the first component is much more significant than the rest of the components (oblique structure). This is the case with image neighborhoods where the first component, which is always approximately the average intensity in the neighborhood, has a much larger eigen value than the rest of the components. Therefore, this algorithm removes the effect of the first component. Here we

$$\mu_i = \frac{1}{M} \sum_{k \in N_i} y_{i,k}$$

compute the average intensity of the neighborhood and generate a

new set of neighborhood vectors whose elements are $y'_{i,k} = y_{i,k} - \mu_i$. Finally, the

artificial data are generated from the permutations $y_{i,k} = y'_{j(i),k}$.

3.4 Smoothing Kernel Width Selection

The optimal choice of the parameter h in equations (5), (6) and (7) varies significantly with the image neighborhood size and choice of subspace dimensionality. We empirically find the optimal h for each combination of d and N for the set of test images used in this paper. This is repeated at various noise standard deviations added to the images. The optimal value of h behaves in a very predictable manner as a function of the noise level and PCA subspace dimensionality .

4 Results and Analysis

In this section, we present detailed experimental results studying the behavior of the proposed algorithm with respect to subspace dimensionality and search-window size selections. We also present quantitative and qualitative comparisons with the original NLM algorithm applied for color images. We study the performance of the proposed approach using images corrupted with additive, independent Gaussian noise with zero mean and standard deviations 10, 15 and 20.

In section 4.1. we present the comparison across various image neighborhood sizes. In section 4.2, we will compare the performance of the full proposed algorithm to the NLM algorithm for color images.

4.1 Subspace Dimensionality and Image Neighborhood Size

Comparison across various image neighborhood sizes are given in Table 1. For each test image, Table 1 includes three rows, one for each input noise level. Each row gives the best PSNR values at the optimal choice of d for different neighborhood sizes. Results for image neighborhoods ranging from 5×5 to 9×9 are provided.. Finally, the overall best PSNR across the various neighborhood sizes for a particular image and noise level is shown in boldface.

Table 1. Psnr values at the optimal subspace dimensionality for three noise levels ($\sigma= 10, 15, 20$) shown in column 3. Columns 4-6 show the results at the best for neighborhood sizes from 5×5 to 9×9 .

Input Image	Noise Variances	PSNR of the noisy image	5×5 NLMP CA	7×7 NLMP CA	9×9 NLMP CA
Lena	10	32.05	34.40	34.45	34.45
	15	30.17	33.04	33.04	33.03
	20	29.30	32.13	32.03	32.13
Baloon	10	32.05	33.47	33.49	33.52
	15	30.21	32.18	32.19	32.17
	20	29.33	31.32	31.45	31.47
Island	10	32.13	32.81	32.80	32.79
	15	30.27	31.34	31.32	31.32
	20	29.38	30.65	30.66	30.61
House	10	32.05	33.81	33.83	33.87
	15	30.18	32.59	32.59	32.61
	20	29.29	31.81	31.85	31.85
Tree	10	32.04	32.34	32.38	32.25
	15	30.19	31.09	31.12	31.18
	20	29.29	30.37	30.35	30.37
Girl	10	32.01	35.31	35.31	35.27
	15	30.17	33.80	33.80	33.80
	20	29.31	33.00	33.00	33.00

As the noise level increases, weight reliability becomes increasingly important; hence, larger image neighborhoods are preferred.

Table 2. PCA subspace dimensionality selected by parallel analysis

Input Image	Noise Variances	5×5 NLMP CA	7×7 NLMP CA	9×9 NLMP CA
Lena	10	3	4	6
	15	3	4	6
	20	3	4	6
Baloon	10	3	4	5
	15	3	4	5
	20	3	4	5
Island	10	2	4	5
	15	2	4	5
	20	2	4	5
House	10	2	4	5
	15	2	4	4
	20	2	4	4
Tree	10	4	6	8
	15	4	6	8
	20	4	6	8
Girl	10	3	6	8
	15	3	6	8
	20	3	6	7

4.2 Comparison with NLM Algorithm for Color Images

Table 2 shows the values selected by parallel analysis for various neighborhood sizes and noise levels. Table 3 compares the results of the proposed algorithm with these automatically chosen and values to the results of the NLM algorithm for color images. Table 4 shows the time taken for denoising using the proposed algorithm and NLM.

Table 3. Psnr values for images denoised with the proposed algorithm and NLM for color images

Input Image	Noise Variances	5×5		7×7		9×9	
		NLMP CA	NLM	NLMP CA	NLM	NLMP CA	NLM
Lena	10	34.38	33.62	34.42	33.61	34.42	33.62
	15	32.31	31.42	32.31	31.42	32.31	31.42
	20	30.96	30.21	30.92	30.20	30.92	30.20
Baloon	10	33.45	33.10	33.18	33.09	33.52	33.09
	15	31.62	31.17	31.42	31.16	31.63	31.16
	20	30.68	30.08	30.14	30.08	30.62	30.08
Island	10	32.82	32.55	32.82	32.56	32.78	32.56
	15	31.26	30.99	31.47	30.98	31.44	30.99
	20	30.38	30.07	30.57	30.07	30.42	30.07
House	10	33.83	33.49	33.83	33.50	33.85	33.50
	15	31.78	31.39	31.81	31.38	32.02	31.39
	20	30.46	30.22	30.79	30.21	30.73	30.20
Tree	10	32.34	32.45	32.38	32.45	32.25	32.45
	15	31.09	30.96	31.12	30.96	31.18	30.95
	20	30.32	30.00	30.35	30.00	30.33	30.00
Girl	10	34.75	33.88	34.78	33.89	34.41	33.87
	15	31.84	31.54	32.20	31.55	31.91	31.53
	20	30.58	30.22	30.97	30.22	30.98	30.22

Table 4. Time for denoising using the proposed algorithm and NLM for color images of noise variance 10. Image neighborhood size is 7×7.

Input Image	Size	Time in seconds	
		NLMPCA	NLM
Lena	256×256×3	42.765	76.328
Baloon	171×256×3	28.266	51.329
Island	192×256×3	31.594	57.375
House	256×256×3	41.984	76.735
Tree	256×256×3	43.391	76.578
Girl	256×256×3	43.454	76.781

5 Conclusion

The proposed algorithm is a variation of the nonlocal means (NLM) image denoising algorithm that uses principal component analysis (PCA) to achieve a higher accuracy while reducing computational load. The accuracy and computational cost of the NLM image denoising algorithm is improved by computing neighborhood similarities, i.e., averaging weights, after a PCA projection to a lower dimensional subspace. We showed that parallel analysis can be used to automatically determine a subspace dimensionality that yields good results. Consequently, neighborhood similarity weights for denoising are computed using distances in this subspace rather than the full space. Then accuracy of NLM and the proposed algorithm are examined with respect to the choice of image neighborhood and search window sizes. Also the proposed algorithm is compared with NLM image denoising algorithm. This approach can also be easily applied to other denoising and segmentation algorithms that use similarity measures based on image neighborhood vectors.

References

- [1] Tasdizen, T.: Senior Member, IEEE, Principal Neighborhood Dictionaries for Nonlocal Means Image Denoising. *IEEE Trans. Image processing* 18(12) (December 2009)
- [2] Mumford, D., Shah, J.: "Optimal approximations by piecewise smooth functions and associated variational problems. *Commun. Pure Appl. Math.* 42(4), 577–685 (1989)
- [3] Perona, P., Malik, J.: "Scale-space edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.* 12(7), 629–639 (1990)
- [4] Rudin, L.I., Osher, S.J., Fatemi, E.: "Nonlinear total variation based noise removal algorithms. *Phys. D* 60, 259–268 (1992)
- [5] Muresan, D.D., Parks, T.W.: Adaptive principal components and image denoising. In: *Proc. Int. Conf. Image Processing*, vol. 1, pp. 101–104 (2003)
- [6] Buades, A., Coll, B., Morel, J.-M.: A non-local algorithm for image denoising. In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 60–65 (2005)
- [7] Kervrann, C., Boulanger, J.: Optimal spatial adaptation for patchbased image denoising. *IEEE Trans. Image Process.* 15(10), 2866–2878 (2006)
- [8] Awate, S.P., Whitaker, R.T.: Unsupervised, information-theoretic, adaptive image filtering for image restoration. *IEEE Trans. Pattern Anal. Mach. Intell.* 28(3), 364–376 (2006)
- [9] Azzabou, N., Paragios, N., Guichard, F.: Image denoising based on adapted dictionary computation. In: *Proc. Int. Conf. Image Processing*, vol. 3, pp. 109–112 (2007)
- [10] Tasdizen, T.: Principal components for non-local means image denoising. In: *Proc. Int. Conf. Image Processing* (2008)
- [11] Orchard, J., Ebrahim, M., Wang, A.: Efficient nonlocal-means denoising using the SVD. In: *Proc. Int. Conf. Image Processing* (2008)
- [12] Efros, A.A., Leung, T.K.: Texture synthesis by non-parametric sampling. In: *Proc. IEEE Int. Conf. Computer Vision*, pp. 1033–1038 (1999)
- [13] Wei, L.-Y., Levoy, M.: Fast texture synthesis using tree-structured vector quantization. In: *Proc. SIGGRAPH*, pp. 479–488 (2000)
- [14] Awate, S.P., Tasdizen, T., Whitaker, R.T.: Unsupervised texture segmentation with non-parametric neighborhood statistics. In: *Proc. Eur. Conf. Computer Vision*, pp. 494–507 (2006)

- [15] Horn, J.L.: A rationale and test for the number of factors in factor analysis. *Psychometrika* 30(2), 179–185 (1965)
- [16] Weickert, J.: *Anisotropic Diffusion in Image Proc.* Verlag, Teubner (1998)
- [17] Chan, T., Shen, J., Vese, L.: “Variational Pde models in image proc. *Notice Amer. Math. Soc.* 50, 14–26 (2003)
- [18] Nordstrom, K.N.: Biased anisotropic diffusion: A unified regularization and diffusion approach to edge detection. *Image Vis. Comput.* 8(4), 318–327 (1990)
- [19] Kindermann, S., Osher, S., Jones, P.W.: Deblurring and denoising of images by nonlocal functionals. *Multiscale Model. Simul.* 4(4), 1091–1115 (2005)
- [20] Brox, T., Kleinschmidt, O., Cremers, D.: Efficient nonlocal means for denoising of textural patterns. *IEEE Trans. Image Processing* 17(7), 1083–1092 (2008)
- [21] Gorfeld, L.W.: An improvement on horn’s parallel analysis methodology for selecting the correct number of factor’s to retain. *Educ. Psych. Meas.* 55(3), 377–393 (1995)
- [22] Starck, J., Candes, E., Donoho, D.: The curvelet transform for image denoising. *IEEE Trans. Image Process.* 11(6) (June 2000)
- [23] Pizurica, A., Philips, W., Lemahieu, I., Acheroy, M.: A joint inter and intrascale statistical model for bayesian wavelet based image denoising. *IEEE Trans. Image Process* 11, 545–557 (2002)
- [24] Sendur, L., Selesnick, I.: Bivariate shrinkage functions for waveletbased denoising exploiting interscale dependency. *IEEE Trans. Signal Process.* 50, 2744–2756 (2002)
- [25] Portilla, J., Strela, V., Wainwright, M., Simoncelli, E.: Image denoising using scale mixtures of gaussians in the wavelet domain. *IEEE Trans. Image Process.* 12, 1338–1351 (2003)
- [26] Elad, M., Aharon, M.: Image denoising via sparse and redundant representations over learned dictionary. *IEEE Trans. Image Process.* 15(12), 3736–3745 (2006)
- [27] Roth, S., Black, M.J.: Fields of experts: A framework for learning image priors with applications. In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 2, pp. 860–867 (2005)
- [28] Mahmoudi, M., Sapiro, G.: Fast image and video denoising via nonlocal means of similar neighborhoods. *IEEE Signal Proc. Lett.* 12(12), 839–842 (2005)

An Iterative Method for Multimodal Biometric Face Recognition Using Speech Signal

M. Nageshkumar* and M.N. ShanmukhaSwamy

Department of Electronics and Communication., J.S.S. Research Foundation,
University of Mysore., Mysore-06
nageshkumar79m@gmail.com, mnsjce@gmail.com

Abstract. In recent years much advancement have been made in face recognition techniques to cater to the challenges such as pose, expression, illumination, aging and disguise. However, due to advances in technology, there are new emerging challenges for which the performance of face recognition systems degrades and plastic/cosmetic surgery is one of them. In this paper we comment on the effect of plastic surgery face image in multimodal biometric face recognition using speech signal. Speaker identity is correlated with the physiological and behavioral characteristics of the speaker. Selecting the most effective fusion techniques depends on operational issues such as accuracy requirements, availability of training data, and the validity of simplifying assumptions.

Keywords: Multimodal biometric system, plastic surgery face image, speech signal and matching level fusion.

1 Introduction

Human biometric characteristics are unique, so it can hardly be duplicated [1]. Such information includes: facial, speech, hands, body, fingerprints, and gesture to name a few. Face detection and recognition techniques are proven to be more popular than other biometric features based on efficiency and convenience [2, 3]. Face authentication has become a potential a research field related to face recognition. Face recognition differs from face authentication because the former has to determine the identity of an object, while the latter needs to verify the claimed identity of a user. Speech [4] is one of the basic communications, which is better than other methods in the sense of efficiency and convenience. Each a single biometric information, however, has its own limitation. For this reason, we present a multimodal biometric verification/identification method to reduce false acceptance rate (FAR) and false rejection rate (FRR) in real-time.

A multimodal biometric face recognition is a well studied problem in which several approaches have been proposed to address the challenges of illumination [9,10], pose [11,12,13], expression [10], aging [14,15] and disguise [16,17], the growing popularity of plastic surgery introduces new challenges in designing future face

* Corresponding author.

recognition systems. Since these procedures modify both the shape and texture of facial features to varying degrees, it is difficult to find the correlation between pre and post surgery facial geometry. Due to the sensitive nature of the process and the privacy issues involved, it is extremely difficult to prepare a face database that contains images before and after surgery. After surgery, the geometric relationship between facial features changes and there is no technique to detect and measure such type of alterations.

The main aim of this paper is to add a new dimension to face recognition by using speech signal and discussing this challenge and systematically evaluating the performance of existing faces recognition algorithms on a database that contains face images before and after surgery.

Related Work

Brunelli and Falavigna [18] used hyperbolic tangent (\tanh) for normalization and weighted geometric average for fusion of voice and face biometrics. Kittler [19,23] have experimented with several fusion techniques for face and voice biometrics, including sum, product, minimum, median, and maximum rules and they have found that the sum rule outperformed others.

Hong and Jain [20] proposed an identification system based on face and fingerprint, where fingerprint matching is applied after pruning the database via face matching. Ben-Yacoub [21, 24, 25] considered several fusion strategies, such as support vector machines, tree classifiers and multi-layer perceptions, for face and voice biometrics. Ross and Jain [22] combined face, fingerprint and hand geometry biometrics with sum, decision tree and linear discriminant-based methods. The authors report that sum rule outperforms others.

The rest of this paper is organized as follows. Section 2 presents the proposed Diagonal PCA method for face feature extraction. Section 3 presents the speech feature extraction method. Also section 4 presents the fusion at matching score level. Section 5 reports on the experimental results. Finally, Section 6 concludes.

2 Face Feature Extraction

Our motivation for developing the Diagonal PCA method originates from an essential observation on the recently proposed 2DPCA [28]. That is, 2DPCA can be seen as the row-based PCA [26, 27], which has been pointed out in [29]. So 2DPCA only reflects the information between rows, which implies some structure information (e.g. regions of a face like eyes, nose, etc.) cannot be uncovered by it. We attempt to solve that problem by transforming the original face images into corresponding *diagonal face images*. Because the rows (columns) in the transformed diagonal face images simultaneously integrate the information of rows and columns in original images, it can reflect both information between rows and those between columns. Through the entanglement of row and column information, it is expected that Diagonal PCA may find some useful block or structure information for recognition in original images. Suppose that there are M training face images, denoted by m by n matrices A_k ($k = 1, 2, \dots, M$).

For each training face image, define the corresponding *diagonal face image* as follows:

- 1) If the height m is equal to or smaller than the width n , use the method illustrated in Figure.1 to generate the diagonal image B for the original image A .

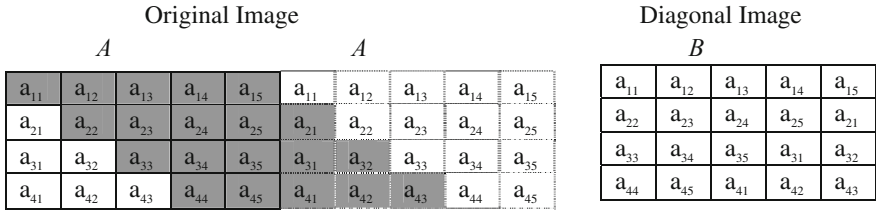


Fig. 1. Illustration for deriving the diagonal face images

Without loss of generalization, assume that the width n is no smaller than the height m . For each training face image A_k , derive the corresponding diagonal face B_k using the method illustrated in Figure.1 Note that B_k is of the same size of A_k .

- 2) If the height m is bigger than the width n , use the method illustrated in Figure.2 to generate the diagonal image B for the original image A .

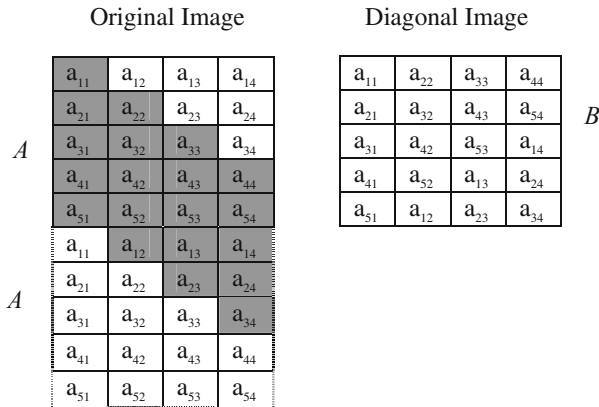


Fig. 2. Illustration for deriving the diagonal face images

Based on the diagonal faces, define the *diagonal covariance matrix* as

$$G = \frac{1}{M} \sum_{k=1}^M (B_k - \bar{B})^T (B_k - \bar{B}) \tag{1}$$

where $\bar{B} = \frac{1}{M} \sum_k B_k$ is the mean diagonal face image, according to eq. (1), the projective vectors X_1, \dots, X_d can be obtained by computing the d eigenvectors corresponding

to the d biggest eigenvalues of G . Since the size of G is only n by n , computing its eigenvectors can be efficient. Let $X=[X_1, \dots, X_d]$ denote the projective matrix, projecting training faces A_k s onto X , yielding m by d feature matrices

$$C_k = A_k X \tag{2}$$

Given a test face image A , first use eq. (2) to get the feature matrix $C=AX$, then a nearest neighbor classifier can be used for classification. Here the distance between C and C_k is defined as:

$$d(C, C_k) = \|C - C_k\| = \sqrt{\sum_{i=1}^m \sum_{j=1}^d (C^{(i,j)} - C_k^{(i,j)})^2} \tag{3}$$

3 Speech Feature Extraction

The objective of voice recognition is to determine which speaker is present based on the individual’s utterance [30]. Several techniques have been proposed for reducing the mismatch between the testing and training environments. Many of these methods operate either in spectral [31, 32], or in cepstral domain [33]. First, the speech samples are processed using MFCC to produce voice features. After that, the coefficient of voice features can go through DTW to select the pattern that matches the database and input frame in order to minimize the resulting error between them.

Text-dependent methods are usually based on template-matching techniques. In this approach, the input utterance is represented by a sequence of feature vectors, generally short-term spectral feature vectors. The time axes of the input utterance and each reference template or reference model of the registered speakers are aligned using a dynamic time warping (DTW) algorithm and the degree of similarity between them, accumulated from the beginning to the end of the utterance, is calculated.

A voice analysis is done after taking an input through microphone from a user. The design of the system involves manipulation of the input audio signal. At different levels, different operations are performed on the input signal such as Pre-emphasis, Framing, Windowing, Mel Cepstrum analysis and Recognition (Matching) of the spoken word.

3.1 Feature Extraction Using MFCC

Step 1: Pre-emphasis

This step processes the passing of signal through a filter which emphasizes higher frequencies. This process will increase the energy of signal at higher frequency.

$$Y[n] = X[n] - 0.95 X[n-1] \tag{4}$$

Lets consider $a = 0.95$, which make 95% of any one sample is presumed to originate from previous sample.

Step 2: Framing

The voice signal is divided into frames of N samples. Adjacent frames are being separated by M ($M < N$). Typical values used are $M = 100$ and $N = 256$.

Step 3: Hamming windowing

Hamming window is used as window shape by considering the next block in feature extraction processing chain and integrates all the closest frequency lines. The Hamming window equation is given as:

If the window is defined as $W(n), 0 \leq n \leq N-1$

Where- N = number of samples in each frame

$Y[n]$ = Output signal

$X(n)$ = input signal

$W(n)$ = Hamming window, then the result of windowing signal is shown below:

$$Y(n) = X(n) \times W(n) \tag{5}$$

$$w(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right) \quad 0 \leq n \leq N-1 \tag{6}$$

Step 4: Fast Fourier Transform

To convert each frame of N samples from time domain into frequency domain. The Fourier Transform is to convert the convolution of the glottal pulse $U[n]$ and the vocal tract impulse response $H[n]$ in the time domain. This statement supports the equation below:

$$Y(w) = FFT [h(t) * X(t)] = H(w) * X(w) \tag{7}$$

If $X(w), H(w)$ and $Y(w)$ are the Fourier Transform of $X(t), H(t)$ and $Y(t)$ respectively.

Step 5: Mel Filter Bank Processing

The frequencies range in FFT spectrum is very wide and voice signal does not follow the linear scale. To compute a weighted sum of filter spectral components so that the output of process approximates to a Mel scale. Each filter’s magnitude frequency response is triangular in shape and equal to unity at the centre frequency and decrease linearly to zero at centre frequency of two adjacent filters [7, 8]. Then, each filter output is the sum of its filtered spectral components. After that the following equation is used to compute the Mel for given frequency f in HZ:

$$F(Mel) = [2595 * \log_{10} [1+f] 700] \tag{8}$$

Feature Matching Using DTW

The principle of DTW is to compare two dynamic patterns and measure its similarity by calculating a minimum distance between them.

Suppose we have two time series P and Q , of length n and m respectively, where:

$$P = p_1, p_2, \dots, p_v, \dots, p_n \tag{9}$$

$$Q = q_1, q_2, \dots, q_v, \dots, q_m \tag{10}$$

To align two sequences using DTW, an n -by- m matrix where the (i^{th}, j^{th}) element of the matrix contains the distance $d(p_i, q_j)$ between the two points p_i and q_j is constructed. Then, the absolute distance between the values of two sequences is calculated using the Euclidean distance computation:

$$d(p_i, q_j) = (p_i - q_j)^2 \tag{11}$$

Each matrix element (i, j) corresponds to the alignment between the points p_i and q_j . Then, accumulated distance is measured by:

$$D(i, j) = \min [D(i-1, j-1), D(i-1, j), D(i, j-1)] + d(i, j) \tag{12}$$

4 Fusion at the Matching Score Level

In the context of verification, there are two approaches for consolidating the scores obtained from different matchers. One approach is to formulate it as a classification problem, where a feature vector is constructed using the matching scores output by the individual matchers; this feature vector is then classified into one of the two classes: ‘‘Genuine user’’ or ‘‘Impostor’’. In the combination approach, the individual matching scores are combined to generate a single scalar score, which is then used to make the final decision. Now consider a multimodal biometric verification system that utilizes the combination approach to fusion at the match score level. The theoretical framework developed by Kittler [34] can be applied to this system only if the output of each modality is of the form $P(\text{genuine}|Z)$ i.e., the posteriori probability of user being ‘‘genuine’’ given the input biometric sample Z . In practice, most biometric systems output a matching score s , and Verlinde [35] have proposed that the matching score s is related to $P(\text{genuine}|Z)$ as follows:

$$S = f \{ P(\text{genuine} | Z) \} + \eta(Z) \tag{13}$$

where f is a monotonic function and η is the error made by the biometric system that depends on the input biometric sample Z . This error could be due to the noise introduced by the sensor during the acquisition of the biometric signal and the errors made by the feature extraction and matching processes. If we assume that η is zero, it is reasonable to approximate $P(\text{genuine}|Z)$ by $P(\text{genuine}|s)$. In this case, the problem reduces to computing $P(\text{genuine}|s)$ and this requires estimating the conditional densities $P(s|\text{genuine})$ and $P(s|\text{impostor})$. The probability of the score being that of a genuine user was then computed as,

$$P(\text{genuine} | s) = \frac{p(s | \text{genuine})}{p(s | \text{genuine}) + p(s | \text{impostor})} \tag{14}$$

5 Experimental Results

To evaluate the performance of face recognition algorithms in such an application scenario, the plastic surgery database is partitioned into two groups: training database and testing database. This partition ensures that the verification is performed on

unseen images. The train-test partitioning is repeated again and again by computing the false rejection rates (FRR) over these trials at different false accept rate (FAR). The verification accuracy is computed at 5.26% FAR. The experimental result for the recognition rate using the proposed method is summarized in Table 1. An experimental result of FAR given in Table 1 corresponds to 5.26%. In this case, the FAR can accept a person out of 120. Table 2 shows the result of the recognition rate and FAR for the proposed method.

Table 1. Verification rates of face and speech

Test Database	Verification Rates (%)	FAR (%)
Face	88.52	11.48
Speech	92.37	7.63

Table 2. Verification rate of the proposed method

Test Database	Verification Rates (%)	FAR (%)
Fusion of Face & Speech	94.74	5.26

6 Conclusion

In this paper, we present a multimodal biometric human identification method using combined plastic surgery face image and speech information in order to improve the problem of multimodal biometric face recognition system. Current face recognition algorithms mainly focus on handling pose, expression, illumination, aging and disguise. This paper formally introduces plastic surgery as another major challenge for face recognition algorithms using speech signal. Based on the results, we believe that more research is required to design optimal face recognition algorithms that can account for the challenges due to plastic surgery. The procedures can significantly change the facial regions both locally and globally, altering the appearance, facial features and texture. Existing face recognition algorithms generally rely on this information and any variation can affect the multimodal biometric recognition performance.

References

1. Kong, S., Heo, J., Abidi, B., Paik, J., Abidi, M.: Recent advances in visual and infrared face recognition - A review. *Computer Vision and Image Understanding* 97(1), 103-135, 1077-3142 (2005)
2. Kriegman, D., Yang, M., Ahuja, N.: Detecting faces in images: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(1), 34-58, 0162-8828 (2002)
3. Liu, X., Chen, T., Kumar, V.: On modeling variations for face authentication. In: *Proceedings of International Conference Automatic Face and Gesture Recognition*, pp. 369-374 (May 2002)

4. Gu, Y., Thomas, T.: A hybrid score measurement for HMM-based speaker verification. In: Proceedings of IEEE International Conference Acoustics, Speech, and Signal Processing, vol. 1, pp. 317–320 (1999)
5. Rowley, H., Baluja, S., Kanade, T.: Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20(1), 203–208, 0162–8828 (1998)
6. Osuna, E., Freund, R., Girosi, F.: Training support vector machines: an application to face detection. In: Proceeding of IEEE Conference Computer Vision and Pattern Recognition, pp. 130–136 (1997)
7. Samaria, F., Young, S.: HMM based architecture for face identification. *Image and Vision Computing* 12(8), 537–543, 262–8856 (1994)
8. Belhumeur, P., Hespanha, J., Kriegman, D.: Eigenfaces vs fisherfaces: recognition using class specification linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19(7), 711–720, 162–8828 (1997)
9. Li, S., Chu, R., Liao, S., Zhang, L.: Illumination invariant face recognition using near-infrared images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29(4), 627–639 (2007)
10. Singh, R., Vatsa, M., Noore, A.: Improving verification accuracy by synthesis of locally enhanced biometric images and deformable model. *Signal Processing* 87(11), 2746–2764 (2007)
11. Blanz, V., Romdhani, S., Vetter, T.: Face identification across different poses and illuminations with a 3d morphable model. In: Proceedings of International Conference on Automatic Face and Gesture Recognition, pp. 202–207 (2002)
12. Liu, X., Chen, T.: Pose-robust face recognition using geometry assisted probabilistic modeling. In: Proceedings of International Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 502–509 (2005)
13. Singh, R., Vatsa, M., Ross, A., Noore, A.: A mosaicing scheme for pose-invariant face recognition. *IEEE Transactions on Systems, Man and Cybernetics - Part B* 37(5), 1212–1225 (2007)
14. Lanitis, A., Taylor, C.J., Cootes, T.F.: Toward automatic simulation of aging effects on face images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(4), 442–450 (2002)
15. Ramanathan, N., Chellappa, R.: Face verification across age progression. *IEEE Transactions on Image Processing* 15(11), 3349–3362 (2006)
16. Ramanathan, N., Chowdhury, A.R., Chellappa, R.: Facial similarity across age, disguise, illumination and pose. In: Proceedings of International Conference on Image Processing, pp. 1999–2002 (2004)
17. Singh, R., Vatsa, M., Noore, A.: Face recognition with disguise and single gallery images. *Image and Vision Computing* 27(3), 245–257 (2009)
18. Brunelli, R., Falavigna, D.: Person Identification Using Multiple Cues. *IEEE Trans. PAMI* 17(10), 955–966 (1995)
19. Kittler, J., Hatef, M., Duin, R.P.W., Matas, J.: On Combining Classifiers. *IEEE Trans. PAMI* 20(3), 226–239 (1998)
20. Hong, L., Jain, A.K.: Integrating Faces and Fingerprints for Personal Identification. *IEEE Trans. PAMI* 20(12), 1295–1307 (1998)
21. Ben-Yacoub, S., Abdeljaoued, Y., Mayoraz, E.: Fusion of Face and Speech Data for Person Identity Verification. *IEEE Trans. N Networks* 10(5) (1999)
22. Ross, A., Jain, A.K.: Information Fusion in Biometrics. *Pattern Recognition Letters* 24(13), 2115–2125 (2003)

23. Kittler, J., Hatef, M., Duin, R., Matas, J.: On Combining Classifiers. *IEEE Trans on Pattern Analysis and Machine Intelligence* 20(3) (March 1998)
24. Ben-Yacoub, S., Abdeljaoued, S., Mayoraz, E.: Fusion of Face and Speech Data for Person Identity Verification (1999)
25. Fierrez-Aguilar, J., Ortega-Garcia, J., Garcia-Romero, D., Gonzalez-Rodriguez, J.: A comparative evaluation of fusion strategies for multimodal biometric verification. In: Kittler, J., Nixon, M.S. (eds.) AVBPA 2003. LNCS, vol. 2688, pp. 830–837. Springer, Heidelberg (2003)
26. Chen, S.C., Zhu, Y.L., Zhang, D.Q., Yang, J.Y.: Feature extraction approaches based on matrix pattern: MatPCA & MatFLDA. *Pattern Recognition Letters* 26(8) (2005)
27. Turk, M., Pentland, A.: Eigenfaces for recog. *J. Cognitive Neuroscience* (1991)
28. Yang, J., Zhang, D., Frangi, A.F., Yang, J.Y.: Two-dimensional PCA: a new approach to appearance-based face representation and recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 26(1), 131–137 (2004)
29. Zhang, D.Q., Chen, S.C., Liu, J.: Representing image matrices: Eigenimages vs. Eigenvectors. In: Wang, J., Liao, X.-F., Yi, Z. (eds.) ISNN 2005. LNCS, vol. 3497, pp. 659–664. Springer, Heidelberg (2005)
30. Yee, C.S., Ahmad, A.M.: Malay Language Text Independent Speaker Verification using NN-MLP classifier with MFCC (2008)
31. Lockwood, P., Boudy, J.: Experiments with a Nonlinear Spectral Subtractor (NSS), Hidden Markov Models and the Projection, for Robust Speech Reco (1992)
32. Rosenberg, A., Lee, C.-H., Soong, F.: Cepstral Channel Normalization Techniques for HMM Based Speaker Verification (1994)
33. Jackson, P.: Features extraction 1.ppt, University of Surrey, GU2 & 7XH
34. Kittler, J., Hatef, M., Duin, R.P., Matas, J.G.: On combining classifiers. *IEEE Trans. Pattern Anal. Mach. Intell.* 20(3), 226–239 (1998)
35. Verlinde, P., Druyts, P., Cholet, G., Achery, M.: Applying Bayes based classifiers for decision fusion in a multimodal identity verification system. In: Proceedings of International Symposium on Pattern Recognition In Memoriam Pierre Devijver, Brussels, Belgium (1999)

Quality Analysis of a Chaotic and Hopping Stream Based Cipher Image

G.A. Sathishkumar^{1,*}, K. Bhoopathybagan², and N. Sriraam³

¹ Assistant Professor, Department of Electronics and Communication Engineering,
Sri Venkateswara College of Engineering, Sriperumbudur- 602108
sathish@svce.ac.in

² Professor and HEAD, Department of Electronics, Madras Institute of Technology,
Chrompet, Chennai- 600044

³ Center for Biomedical Informatics and Signal Processing,
Department of Biomedical Engineering
SSN College of Engineering, Chennai- 603110

Abstract. This paper is devoted to the analysis of the impact of chaos-based techniques on stream ciphers. The chaos based cryptographic algorithm has suggested a new and efficient ways to develop secure image encryption technique. In this paper, the keys are generated by four different chaotic maps. Based on the initial conditions, each map may produce various random numbers from various orbits of the maps. Among those random numbers, a particular number and from a particular orbit are selected as a key for the encryption algorithm. Based on the key, a binary sequence is generated to control the encryption algorithm. The input image of 2-D is transformed into a 1- D array by using scanning pattern and then divided into various sub blocks. Then the position permutation and value permutation is applied to each binary matrix based on chaos maps. Cryptanalysis proves that among various chaos maps Bernoulli map is more resistant to known text and cipher text only attack.

Keywords: Logistic Map, Tent Map, Quadratic Map, and Bernoulli Map, Chaos, Hopping, Stream Cipher.

1 Introduction

In recent years, more and more consumer electronic services and devices, such as mobile phones and PDA (personal digital assistant), have also started to provide additional functions of saving and exchanging multimedia messages [10], [11], [13]. The prevalence of multimedia technology in our society has promoted digital images and videos to play a more significant role than the traditional dull texts, which demands a serious protection of users' privacy. To fulfill such security and privacy needs in various applications, encryption of images and videos is very important to frustrate malicious attacks from unauthorized parties. Due to the tight relationship between

* Corresponding author.

chaos theory[14],[15] and cryptography, chaotic cryptography have been extended to design image and video encryption schemes

1.1 The Need for Image Encryption Schemes

The simplest way to encrypt an image or a video is perhaps to consider the 2-D and 3-D stream as a 1-D data stream, and then encrypt this 1-D stream with any available key , such a simple idea of encryption is called naive encryption[7],[20]. Although naive encryption is sufficient to protect digital images and videos in some civil applications, this issues have taken into consideration when advanced encryption algorithms are specially designed for sensitive digital images and videos, for their special features are very different from texts.

The recent research activities in the field of nonlinear dynamics and especially on systems with complex (chaotic) behaviors [3], [14] have forced many investigations on possible applications of such systems. Today, chaotic encryption [4],[5],[6],[7]is almost exclusively considered inside the nonlinear systems community.

1.2 Chaotic Maps

Chaos theory [3], [14], [15] describes the behavior of certain nonlinear dynamic system that under specific conditions exhibit dynamics that are sensitive to initial conditions. The two basic properties of chaotic systems are the sensitivity to initial conditions and Mixing Property. In this paper, 1 D [15] chaotic map is used to produce the chaotic sequence and used to control the encryption process. The chaos streams are generated by using various chaotic maps. Among the various maps, four maps are investigated and their characteristics are analyzed.

Logistic Map

A simple and well-studied example [3], [14] of a 1D map that exhibits complicated behavior is the logistic map from the interval [0,1] in to [0,1] , parameterised by μ :

$$g_{\mu}(x) = \mu * (x) \tag{1}$$

The state evolution is described by $x(n+1)=\mu*x(n)*(1-x(n))$ (2)

Where $0 \leq \mu \leq 4$. This map constitutes a discrete-time dynamical system in the sense that the map $g_{\mu} : [0,1] \rightarrow [0,1]$ generates a semi-group through the operation of composition of functions. In the logistic map, as μ is varied from 0 to 4, a period-doubling bifurcation occurs.

Tent Map

In mathematics, the tent map [3],[14] is an iterated function, in the shape of a tent, forming a discrete-time dynamical system. It takes a point x_n on the real line and maps it to another point:

$$x_{n+1} = \begin{cases} \mu x_n & \text{for } x_n < \frac{1}{2} \\ \mu(1 - x_n) & \text{for } \frac{1}{2} \leq x_n. \end{cases} \tag{3}$$

Where μ is a positive real constant.

Depending on the value of μ , the tent map demonstrates a range of dynamical behavior ranging from predictable to chaotic.

Quadratic Map

More complicated analytic quadratic map [3],[14] is

$$x_{n+1} = f_c(x_n) = x_n^2 + c \quad (4)$$

For an analytic map points where $f'(x_c) = 0$ are called critical points. Quadratic map has the only critical point $x_c = 0$. So a fixed point is stable (attracting), super stable, repelling, indifferent (neutral) according as its multiplier satisfies $|m| < 1$, $|m| = 0$, $|m| > 1$ or $|m| = 1$. The second fixed point is always repelling. For $|x| > x_2$ iterations go to infinity. For $|x| < x_2$ they go to the attracting fixed point x_1 . This interval is the basis of attraction of the point.

Bernoulli Map

Bernoulli map [3],[14] or the $2x \bmod 1$ map defined as

$$f(x) = \begin{cases} 2x, & 0 \leq x < 0.5 \\ 2x-1, & 0.5 \leq x < 1 \end{cases} \quad (5)$$

A Bernoulli process is a discrete time stochastic process consisting of a finite or infinite sequence of independent random variable X_1, X_2, X_3, \dots , such that for each i , the value of X_i is either 0 or 1; for all values of i , the probability that $X_i = 1$ is the same number p . From any given time, future trials are also a Bernoulli process independent of the past trails.

2 Multi Map Orbit Hopping

In this paper, the chaotic key generation is shown in block diagram [19] Figure 1. Given a key, the hopping mechanism performs a key handling process, then chooses m maps $M0, M1, \dots, Mm-1$ from the chaotic map bank and sets the order of the chosen maps to hop. In addition, for each individual chosen map, s orbits $S0, S1, \dots, Ss-1$ are generated. Further, on each orbit, n points $N0, N1, \dots, Nn-1$ are generated. The key determines parameters m, s, n , and hopping pattern. For a given chaotic map, the second orbit is generated by increasing the initial seed of the first orbit by an offset.

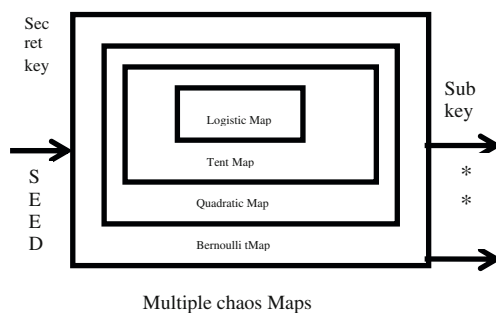


Fig. 1. Typical architecture of Multi-Map Orbit Hopping Chaotic Key Generation

The third orbit is generated by increasing the initial seed of the second orbit by the same offset, and so on so forth for other extra orbits needed. In this paper, 1 D chaotic map is used to produce the chaotic sequence and used to control the encryption process. The maps used in this paper [3],[14],[15] are logistic map, Bernoulli map, Tent map and Quadratic map for key generation. The chaos streams are generated by using various chaotic maps.

3 The Proposed Image Security System

The proposed encryption algorithm belongs to the category of the combination of value transformation and position permutation [7]. In this paper, two different types of scanning methods are used and their performances are analyzed.

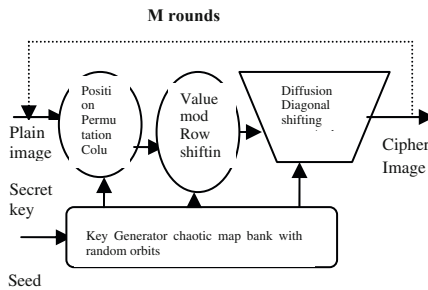


Fig. 2. Proposed Chaos based image cryptosystem

The images are treated as a 1D array by performing Raster scanning and Zigzag scanning. The typical schematic of the proposed method is shown in Figure 2. The scanned arrays are divided into various sub blocks. Then for each sub block, position permutation and value transformation are performed to produce the encrypted image. The sub keys are generated by applying the suitable chaotic map banks. Based on the initial conditions, the generated chaotic map banks are allowed to hop through various orbits of chaotic maps. The hopping pattern [19] is determined from the output of the previous map. Hence, for each sub block various chaotic mapping patterns are applied which further increases the efficiency of the key to be determined by the brute force attack. In each orbit, a sample point is taken and used as key for a specific block and a condition to choose the particular orbit in a particular map is adopted. Then, based on the chaotic system, binary sequence is generated to control the bit-circulation functions for performing the successive data transformation on the input data.

Eight 8-bit data elements are regarded as a set and fed into an 8×8 binary matrix. In the successive transformation on each diagonal by using these two functions, we randomly determine the two parameters used in the functions according to the generated chaotic binary sequence such that the signal could be transformed into completely disorderly data. In addition to chaotic features of mixing, unpredictable, and extreme sensitive to initial seeds, through chaotic maps and orbits hopping mechanism, we spread out the pseudo random number base to a wide flat spread spectrum in terms of time and space. The following steps carried out for the implementation [5][7][19] of proposed chaos based mapping technique.

Let s denote a one-dimensional (1D) [5]digital signal of length N , $s(n)$, $0 \leq n \leq N - 1$, be the one-byte value of the signal s at n , M an 8×8 binary matrix, and s' and M' the encryption results of s and M , respectively. In the following definitions, the integer parameters r and s are assumed larger than or equal to 0, but they are less than 8.

Algorithm

Step1: Covert 2-D image into 1-D array then perform a) Raster scanning and b) Zigzag scanning.

Step2: Consider a block size of 8×8 and convert them in to binary values.

Step3: Sub key size are at least 20 bits, it is extracted from the chaos map banks. The Secret key is SEED, which are the initial conditions of the each map. Based on the initial conditions the chaotic map banks are allowed to hop through various orbits of different chaotic maps. The hopping pattern is determined from the output of the previous map. Then, based on the chaotic system, binary sequence generated to control the bit-circulation functions for performing the successive data transformation on the input data. Given pair of f and f' , the combination of p, q, r, t, u and s resulting in the transformation pair may be non-unique which is the secret key.

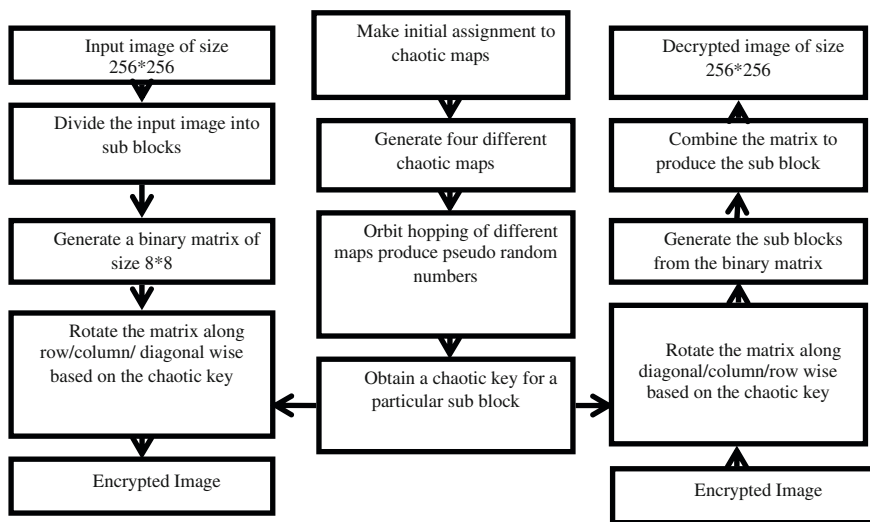


Fig. 3. Typical architecture of the proposed chaos based image crypto systems

Step4: Convert the chaotic sub key in to binary values of 20 bits.

Step5: Each 8×8 -sub block of image pixel values circularly shifted with chaos map banks.

Step6: **Definition for 2 D Circular Shifting of Diagonal pixels (CSDP):** The Mapping [5] $ROLR_k^{t,u}$ & $ROD_k^{t,u} f \rightarrow f'$ is defined to rotate each pixel at the position

(x, y) in the image such that k^{th} diagonal of f $0 \leq f \leq u$ bits in the up direction if t

equals 1 or u bits in the down direction if t equals 0. In different combinations of p, q, r, t, u and s, the composite mapping is given below

$$(\sum_{j=0}^7 ROLR_j^{q,s}) (\sum_{i=0}^7 ROLR_i^{p,r}) . (\sum_{k=0}^{13} ROLR_k^{t,u}) \tag{6}$$

The proposed method (**CSDP**) possesses the following desirable features:

A binary matrix f be transformed into quite different matrixes and different matrixes can be transformed into the same matrix. Given a transformation of pair f and f' the combinations of p, q, r and s resulting in the transformation pair may be non-unique.

Since f is an 8×8 matrix, the result of circulating diagonal k bits and of circulating it (kmod8) bits in the same direction. This is why r and s are assumed to be in the ranges of $0 \leq r \leq 7$ and $0 \leq s \leq 7$.

Step 7: Perform the encryption based on the chaotic key values, which is obtained from the different orbits of chaos maps chosen by hopping randomly.

Step 8: Transform the encrypted image 1-D to 2-D.

Step 9: Transmit the Chaotic sub key via secure channel using public key algorithms.

Step10: Decrypt the cipher image using the same chaotic sub key and SEED.

Finally, carry out performance analysis by doing correlation, histogram, speed and loss of the original, encrypted and decrypted image.

3.1 Analysis of Security Problem for a Mutli – map Hopping

For an unknown set of μ and $x(0)$ of the logistic map, the number of possible encryption results is $2^{16}N/8$ if the TDCEA [5] is applied to a signal of length N. Since it requires $16N/8$ bits to encrypt a signal of length N, the number of possible encryption results is $2^{16}N/8$. Since the chaotic binary sequence is unpredictable and furthermore, proposed technique (**CSDP**) multi –hopping chaotic sequence are used therefore, it is very difficult to decrypt correctly an encrypted signal by making an exhaustive search without knowing μ and $x(0)$. Moreover, small fluctuation in μ and $x(0)$ results in quite different chaotic binary sequence because the trajectory of the chaotic system is very sensitive to initial condition.

By the way of collecting, some original signals and their encryption results or collecting some specified signals and their corresponding encryption results, it is impossible for the crypt analyst to decrypt correctly an encrypted image without knowing μ and $x(0)$. Because the rotation direction and the shifted bit-number in each row or column transformation randomly determined by the multi - hopping chaotic binary sequence. Hence, the new scheme (**CSDP**) can resist the chosen cipher text attack and the known plaintext attack.

4 Experimental Results

An image size of $256 * 256$ (e.g. cameraman, pepper, aero, etc..) is considered as plain (original) image and CSDP is performed with multi map orbit key. The most

direct method to decide the disorderly degree of the encrypted image is by the sense of sight. On the other hand, the correlation coefficient can provide the quantitative measure on the randomness of the encrypted images.

In order to apply the (CSDP), the parameters α and β must be determined according to Step 1. Based on the experimental experience, general combinations of α and β can always result in very disorderly results. In the simulation, $\alpha = 2$ and $\beta = 2$ are adopted in Step 1. The initial conditions of all chaotic maps used are set as, $x(0) = 0.75$ and $\mu = 3.9$ for logistic map, $c=1.75$ for tent map, $f(x)=0.5$ for Bernoulli map and finally $a=.5$, $b=.25$ for quadratic map. The offset values for producing various orbits are chosen to be very less than the initial conditions.

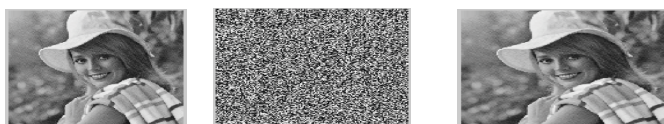


Fig. 4. (a) Original image (b) Encrypted image (c) Decrypted image

The visual inspection of Figure 4 shows the possibility of applying the algorithm successfully in both encryption and decryption. In addition, it reveals its effectiveness in hiding the information contained in it.

4.1 Histogram Analysis

To prevent the leakage of information to an opponent [2],[4],[6] it is also advantageous if the cipher image bears little or no statistical similarity to the plain image.

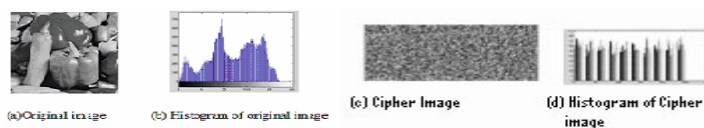


Fig. 5. Histogram of original and encrypted image

We have calculated and analyzed the histograms of the several encrypted images as well as its original images that have widely different content. One typical example among them is shown in Figure 5(b). The histogram of a plain image contains large spikes. The histogram of the cipher image is shown in Figure 5(d), it is uniform, significantly different from that of the original image, and bears no statistical resemblance to the plain image and hence does not provide any clue to employ any statistical attack on the proposed image encryption procedure.

4.2 Correlation Co-efficient Analysis

In addition to the histogram analysis [4],[6], we have also analyzed the correlation between two vertically adjacent pixels, two horizontally adjacent pixels and two

diagonally adjacent pixels in plain image and cipher image respectively. The procedure is as follows:

$$r_{x,y} = \frac{\text{cov}(x,y)}{\sqrt{D(x)}\sqrt{D(y)}} \tag{7}$$

Where x and y are the values of two adjacent pixels in the image.

Figure 6 shows the correlation distribution of two horizontally adjacent pixels in plain image and cipher image.

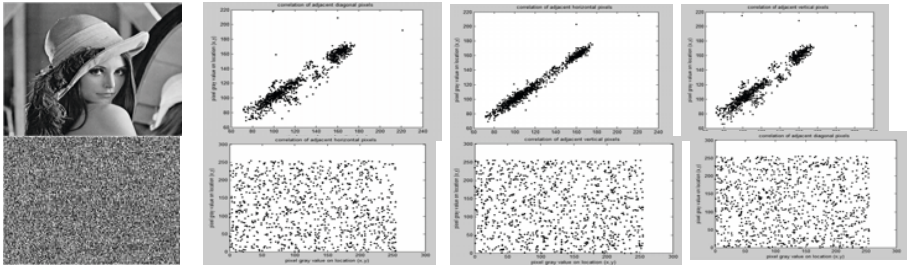


Fig. 6. Horizontal, vertical and diagonal correlation of plain and cipher image

The correlation coefficients are 0.9905 and 0.0308 respectively for both plain image and cipher image. Similar results for diagonal and vertical directions. It is clear from the Figure.6 and Table 1 that there is negligible correlation between the two adjacent pixels in the cipher image. However, the two adjacent pixels in the plain image are highly correlated. The correlation coefficients of various maps are calculated and they are compared with each other. The comparison table for various plain images, various cipher images and various maps based on the correlation coefficient are calculated and tabulated in Table 1 and 2.

Table 1. Horizontal, Vertical & Diagonal Correlation of Encrypted Image

Original Image	Horizontal Correlation	Vertical correlation	Diagonal Correlation
Elaine	-0.0139	-0.0954	-0.0710
Barbara	0.0286	0.0230	-0.0244
Pepper	-0.0025	-0.0219	0.0167
Lena	-0.0590	-0.0381	-0.0457
Cam-eraman	0.0286	0.0230	-0.0244
Airfield	0.0075	0.0069	-0.0317

The correlation coefficients are found for the various directions of scanning patterns employed and the tabulated in the Table 3. In Table 4, the observation shows that the zigzag scanning is more efficient than the raster scanning (See Fig.7). In addition, cipher image with multiple maps are more resistant to crypt analyst attacks.

Table 2. Horizontal Correlation coefficient for various different maps

Images	Logistic Map	Bernoulli Map	Tent Map	Quadratic Map
Elaine	0.0085	-0.2035	-0.0190	-0.2027
Lena	-0.0695	-0.2835	-0.0259	-0.2427
Bridge	0.0061	-0.2349	-0.0327	-0.2357
Airfield	-0.0064	-0.1428	-0.0267	-0.1406

Table 3. Correlation Coefficients in Plain image and Cipher image

Direction of Adjacent Pixels	Plain image	Cipher image using single map	Cipher image using multiple map
Horizontal	0.9670	0.0781	-0.0025
Vertical	0.9870	0.0785	-0.0218
Diagonal	0.9692	0.0683	0.0167

Table 4. Horizontal Correlation coefficients for raster scanning and zigzag scanning

IMAGE	Raster Scanning	Zigzag Scanning
Elaine	0.0539	-0.0139
Lena	-0.0535	-0.0590
Bridge	0.0174	-0.0023
Airfield	-0.0213	0.0075
Barbara	0.0901	0.0286
Peppers	0.0901	-0.0025

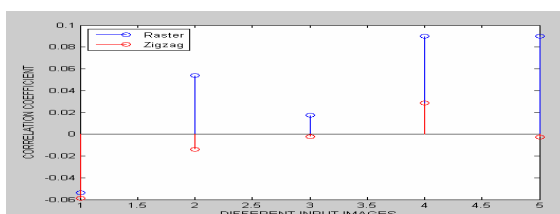
**Fig. 7.** Scanning Patterns**Table 6.** Encryption speed

Image	Size	Speed (sec's)	PSNR
Elaine	256x256	0.1143	9.2637
	64x64	7.3168	
	8x8	117.069540	
Lena	256x256	0.1293	9.2006
	64x64	8.2722	
	8x8	132.354654	
cameraman	256x256	0.1154	8.3399
	64x64	7.3865	
	8x8	118.183788	

5 Conclusion and Future Scope

In addition to chaotic features of mixing, unpredictable, and extreme sensitive to initial seeds, through multiple chaotic maps and orbits hopping mechanism, we spread out the pseudo random number base to a wide flat spread spectrum in terms of time and space. It is similar to say that our pseudo random numbers are out of the white noise. Chaotic maps are computationally economic and fast. This proposed image cipher will be suitable for applications like wireless communications. In future, the proposed crypto system has been implemented and tested in the FPGA hardware.

References

1. Wang., S., Zheng., D., Zhao., J., Tam., W.J., Speranza, F.: An Image Quality Evaluation Method Based on Digital Watermarking. *Transactions Letters IEEE Transactions on Circuits And Systems For Video Technology* 17(1) (2007)
2. El-Fishawy1, N., Osama, M., Zaid2., A.: Quality of encryption Measurement of Bitmap Images with RC6, MRC6, and Rijndael Block Cipher Algorithms. *International Journal of Network Security* 5(3), 241–251 (2007)
3. Dachsel., F., Schwarz, W.: Chaos And Cryptography. *IEEE Transactions on Circuits And Systems—I: Fundamental Theory And Applications* 48, 1498–1501 (2001)
4. Pareek, N.K., Patida, V., Sud, K.K.: Image encryption using chaotic logistic map. *Image and Vision Computing* 24, 926–934 (2006)
5. Chen, H.-C., Guo, J.-I., Huang, L.-C., Yen, J.-C.: Design and Realization of a New Signal Security System for Multimedia Data Transmission. *EURASIP Journal on Applied Signal Processing* 13, 1291–1305 (2003)
6. Liu1, S., Sun, J., Xu1, Z.: An Improved Image Encryption Algorithm based on Chaotic System. *Journal of Computers* 4(11) (2009)
7. Yen, C., Guo, J.-I.: An efficient hierarchical chaotic image encryption algorithm and its VLSI realization. *IEE Proceedings—Vision, Image and Signal Processing* 147(2), 167–175 (2000)
8. Öztürk, S., Soukpinar, Ø.: Analysis and Comparison of Image Encryption Algorithms. In: *Proceedings Of World Academy Of Science, Engineering And Technology*, vol. 3 (2005)
9. Van De Ville, D., Philips, W., Van de Walle, R., Lemahieu, I.: Image scrambling without bandwidth expansion. *IEEE Transactions Circuits and Systems for Video Technology* 14, 892–897 (2004)
10. Yang., M., Bourbakis, N., Shujun, L.: Data-image-video encryption. *Potentials IEEE* 23, 28–34 (2004)
11. Yi, C.H., Tan, C.K., Siew, R., Syed, R.: Fast encryption for multimedia. *IEEE Transactions on Consumer Electronics* 47(1), 101–107 (2001)
12. Kuo, J., Chen, M.S.: A new signal encryption technique and its attack study. In: *Proc. IEEE International Carnahan Conference On Security Technology*, Taipei, Taiwan, pp. 149–153 (1991)
13. Macq, M., Quisquater, J.-J.: Cryptology for digital TV broadcasting. *Proceedings of the IEEE* 83(6), 944–957 (1995)
14. Parker, S., Chua, L.O.: Chaos: a tutorial for engineers. *Proceedings of the IEEE* 75(8), 982–1008 (1995)
15. Wu, W., Rulkov, N.F.: Studying chaos via 1-Dmaps—a tutorial. *IEEE Trans. on Circuits and Systems I: Fundamental Theory and Applications* 40(10), 707–721 (1993)

16. Biham, E.: Cryptanalysis of the chaotic-map cryptosystem suggested at EUROCRYPT'91. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 532–534. Springer, Heidelberg (1991)
17. Wolter, S., Matz, H., Schubert, A., Laur, R.: On the VLSI implementation of the international data encryption algorithm IDEA. In: Proc. IEEE Int. Symp. Circuits and Systems, Seattle, Washington, USA, vol. 1, pp. 397–400 (1995)
18. Kuo, C.J., Chen, M.S.: A new signal encryption technique and its attack study'. In: Proceedings of IEEE international Conference on Security Technology, Taipei, Taiwan, DD, pp. 149–153 (1991)
19. Zhang¹, X., Shu², L., Tang¹, K.: Multi-Map Orbit Hopping Chaotic Stream Cipher. arxiv.org/pdf/cs/0601010-Cornell University Library (2006)
20. Smid, E., Branstad, D.K.: The data encryption standard: past and future. Proceedings of the IEEE 76(5), 550–559 (1988)

Authors



G.A. Sathishkumar obtained his M.E from PSG college of Technology, Coimbatore, India. He is currently perusing PhD from Anna University, Chennai and Faculty member in the Department of Electronics and Communications Sri Venakesateswara College of Engineering, Sripurambudur. His research interest is Image Processing, Network Security, VLSI & Signal processing Algorithms.

Dr. K. Bhoopathy Bagan completed his doctoral degree from IIT Madras. He is presently working as professor, ECE dept, in Anna University, MIT Chrompet campus, Chennai. His areas of interest include Image Processing, VLSI, Signal processing and network Security.

Dr. N. Sriraam He is presently working as Professor and Head Department of Biomedical Engineering SSN College of Engineering, Chennai 603110.

Design Pattern Mining Using State Space Representation of Graph Matching

Manjari Gupta, Rajwant Singh Rao, Akshara Pande, and A.K. Tripathi

DST-Centre for Interdisciplinary Mathematical Sciences,
Department of Computer Science, Banaras Hindu University,
Varanasi, India

manjari_gupta@rediffmail.com, rajwantrao@gmail.com,
pandekshara@gmail.com, aktripathi.cse@itbhu.ac.in

Abstract. Design Pattern Detection is a part of many solutions to Software Engineering problems. It is a part of reengineering process and thus gives significant information to the designer. Design Pattern improves the program understanding and software maintenance. Therefore, a reliable design pattern discovery is required. Graph theoretic approaches have been used for design pattern detection in past. Here we are applying state space representation of graph matching algorithm for design pattern detection. State space representation easily describes the graph matching process. Using our approach variants of each design pattern as well as any occurrence of a design pattern can be detected.

Keywords: design pattern, UML, state space representation, subgraph isomorphism.

1 Introduction

Design Patterns are defined as explanation of corresponding classes that forms a common solution to frequent design problem. The design patterns have been extensively used by software industry to reuse the design knowledge [1]. The availability of pattern related information is generally lost when they are implemented in a system. Therefore, it is tough to trace out such design information. To understand and modify a software system, it is necessary to discover pattern instances in it, if any. Many algorithms have been proposed for design patterns detection like [2], [3], [4], [8]. Similar works on design pattern detection are discussed in section 5.

Here we are proposing a design pattern detection technique using a graph matching algorithm for directed graphs. Here, the directed graphs are basically the relationship graphs which exist in the UML diagrams of system design (model graph or system under study) as well as in UML diagrams of design patterns. The graph matching algorithm that determines whether a graph is sub isomorphic to another graph or not is used here. The main advantage of this algorithm is that the memory requirement is quite lower than from other similar algorithms [5]. In section 2, representation of system design and design patterns in terms of relationship graphs is explained. The graph matching algorithm is described in section 3. In section 4, design pattern detection is

described using some examples. Related works are discussed in section 5. Lastly we concluded in section 6.

2 Relationship Graphs Representation

Firstly, relationships that exist between the classes of the UML diagram of model graph and design pattern graph have been extracted. We have taken the UML Diagram of model graph as shown in Fig. 1. There are four relationships exist in the model graph (i.e. generalization, direct association, aggregation and dependency), the corresponding relationship graphs (i.e. directed graph) are shown in Fig. 2, Fig. 3, Fig. 4, Fig. 5. Generalization relationship graph (Fig. 2) has relationship between only three of the nodes, direct association relationship graph (Fig. 3) has relationship between 4 of the nodes, aggregation relationship graph (Fig. 4) has relationship between 2 of the nodes and dependency relationship graph (Fig. 5) has relationship between 2 of the nodes.

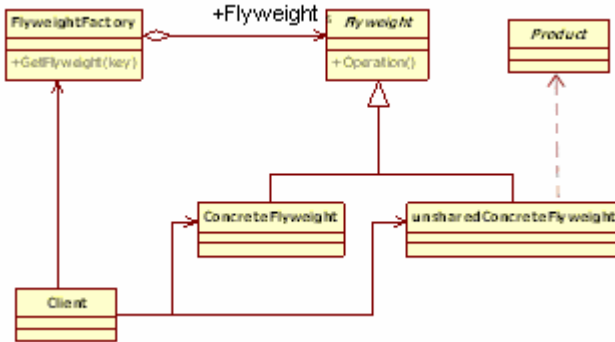


Fig. 1. UML Diagram of model graph

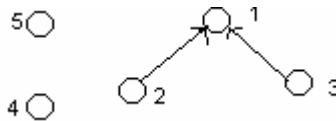


Fig. 2. Generalization Relationship Graph of UML Diagram of model graph

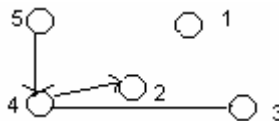


Fig. 3. Direct Association Relationship Graph of UML Diagram of model graph

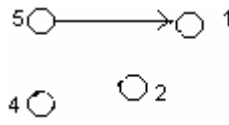


Fig. 4. Aggregation Relationship Graph of UML Diagram of model graph

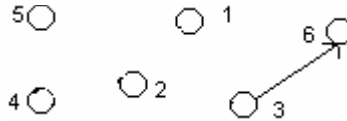


Fig. 5. Dependency Relationship Graph of UML Diagram of model graph

3 Graph Matching Algorithm

The graph matching algorithm [5] determines whether a graph is sub isomorphic to another graph. In this paper we are applying the same algorithm on model graph and design patterns for design pattern detection. Therefore, the following algorithm will be applied on each relationship graphs extracted from UML diagram of both the model graph and design pattern graphs.

Consider a relationship graph of model graph $MG = (N_1, E_1)$ with vertices N_1 and edges E_1 and the same relationship graph of design pattern $DPG = (N_2, E_2)$ with vertices N_2 and edges E_2 , a mapping $M \subset N_1 \times N_2$: MG to DPG is said to be an isomorphism iff it is bijective mapping. M is said to be a graph-sub graph isomorphism iff M is an isomorphism between the relationship graph of DPG and a sub graph of relationship graph of MG. The matching process can be suitably described by means of a State Space Representation (SSR) [5]. Each state s of the matching process can be associated to a partial mapping solution $Match_DPG_MG(s)$, which contains only a subset of the components of the mapping function M . A partial mapping solution univocally identifies two sub graphs of MG and DPG, say $MG(s)$ and $DPG(s)$, obtained by selecting from MG and DPG only the nodes included in the components of $Match_DPG_MG(s)$, and the branches connecting them. In the following, we will denote the projection of $Match_DPG_MG(s)$ onto N_1 and N_2 by $M_1(s)$ and $M_2(s)$ respectively, while the sets of the branches of $MG(s)$ and $DPG(s)$ will be denoted by $B_1(s)$ and $B_2(s)$ respectively. The matching algorithm [5], $Match_DPG_MG(s)$, is given below.

PROCEDURE $Match_DPG_MG(s)$

INPUT: an intermediate state s ; The initial state s_0 has $M(s_0) = \in$

OUTPUT: the mappings between the two graphs

IF $M(s)$ covers all the nodes of DPG relationship graph THEN

 OUTPUT $M(s)$

ELSE

 Compute the set $P(s)$ of the pairs candidate for inclusion in $M(s)$

 FOREACH $(n, m) \in P(s)$

```

IF  $F(s, n, m)$  THEN
  Compute the state  $s'$  obtained by adding  $(n, m)$  to  $M(s)$ 
  CALL Match_DPG_MG ( $s'$ )
END IF
END FOREACH
Restore data structures
END IF
END PROCEDURE

```

Where $F(s, n, m)$ is a Boolean function (called *feasibility function*) that is used to prune the search tree [5]. By applying Match_DPG_MG(s) the largest sub graph match of a particular relationship graph of design pattern in the same relationship graph of model graph can be found.

Given a graph G (can be for design pattern or for model graph) and one of its nodes n , Pred (G, n) (the *predecessors* of n) the set of nodes of G from which a branch originates that ends in n . Similarly, Succ (G, n) (the *successors* of n) the set of nodes of G that are the destination of a branch starting from n . *Out-terminal set* $T^{\text{out}}_1(s)$ is defined as the set of nodes of MG that are not in $M_1(s)$ but are successors of a node in $M_1(s)$, and the *in-terminal set* $T^{\text{in}}_1(s)$ is defined as the set of nodes that are not in $M_1(s)$ but are predecessors of a node in $M_1(s)$. Similarly, $T^{\text{out}}_2(s)$ and $T^{\text{in}}_2(s)$ can also be defined. The set $P(s)$ can be computed as follows [5]:

I). If both $T^{\text{out}}_1(s)$ and $T^{\text{out}}_2(s)$ are not empty, then

$$P(s) = T^{\text{out}}_1(s) \times \{\min T^{\text{out}}_2(s)\} \quad (1)$$

II). If both $T^{\text{out}}_1(s)$ and $T^{\text{out}}_2(s)$ are empty and both $T^{\text{in}}_1(s)$ and $T^{\text{in}}_2(s)$ are not empty, then

$$P(s) = T^{\text{in}}_1(s) \times \{\min T^{\text{in}}_2(s)\} \quad (2)$$

III). If all the four terminal sets are empty, then

$$P(s) = (N_1 - M_1(s) \times \{\min (N_2 - M_2(s))\}) \quad (3)$$

If only one of the in-terminal sets or only one of the out-terminal sets is empty, then the state s cannot be part of a matching, and it is not further explored. To evaluate $F(s, n, m)$ the algorithm examines all the nodes connected to n and m ; if such nodes are in the current partial mapping (i.e. they are in $M_1(s)$ and $M_2(s)$), the algorithm checks if each branch from or to n has a corresponding branch from or to m and vice versa. Otherwise, the algorithm counts how many nodes are in $T^{\text{in}}_1(s)$, $T^{\text{out}}_1(s)$ and $(N_1 - M_1(s) - T^{\text{in}}_1(s) - T^{\text{out}}_1(s))$; for the isomorphism these counts must be equals for n and m , while for the graph-sub graph isomorphism, the count relative to the small graph must be less than or equal to the count for the large graph [5].

4 Design Pattern Detection Using Graph-Matching Algorithm

There are 23 GoF (Gang of Four) [1] design patterns. UML diagrams can be drawn for each of the corresponding design patterns. Here we are considering some of them.

After checking sub isomorphism between all the relationships graphs of a design pattern and the corresponding relationships graphs of the model graph separately, there may be three cases:

- i) Each relationship graph of a design pattern is sub isomorphic to corresponding relationship graph of the model graph.
- ii) Some of these relationship graphs of a design pattern are sub isomorphic to corresponding relationship graph of the model graph.
- iii) No relationship graph of a design pattern is sub isomorphic to corresponding relationship graph of the model graph.

In the case i) design pattern graph may or may not be sub isomorphic to model graph that is design pattern may or may not exist in system design. In the case ii) design pattern partially exists in the system design. In the case iii) design pattern does not exist in the system design. It is necessary to explore the case i). In case i), If for all relationships the pair of nodes (d_i, m_j) , where $d_i \in \text{DPG}$ and $m_j \in \text{MG}$ and d_i is isomorphic to m_j , are matched we say design pattern exist in the system design. But if for one relationship say direct association a pair of nodes (d_i, m_j) does not match with the pair of nodes (d_i, m_k) for another relationship say aggregation then design pattern does not exist in system design.

The algorithm described in section 3 detects sub isomorphism for each relationship separately. Thus, to find out whether a design pattern exists (complete/partially) or not we need to combine result of this algorithm. Here we are introducing a new algorithm that will identify complete sub isomorphism between design pattern graph and model graph by combining the outputs (mappings) of previous algorithm applied separately for all the relationship exists in a design pattern. Firstly, assign unique natural numbers i to each of the relationships of UML and let P_M_i be the corresponding mapping (output M) obtained by the above algorithm for the same relationship.

```

PROCEDURE Union_Relationships
  FOREACH set of mappings in  $P\_M_1$ 
     $F\_M_i = i^{\text{th}}$  mapping in  $P\_M_1$  (  $i$  starts from 1)
    FOREACH mapping in  $P\_M_i$  (in ascending order)  $i \neq 1$ 
       $P\_M_{ij} = j^{\text{th}}$  mapping in  $P\_M_i$  (in ascending order starting from 1)
      IF any pair of  $F\_M_i$  is equal to
        any pair of  $P\_M_{ij}$  and there are
        corresponding edges between all the nodes
           $F\_M_i = F\_M_i \cup P\_M_{ij}$ 
        END IF
      END FOREACH
    END FOREACH
  END FOREACH
END PROCEDURE

```

The above algorithm combines the mappings (outputs), of all relationship graphs of a design pattern to the corresponding relationship graphs of model graph, obtained by the previous algorithm and gives the final mapping (F_M) of design pattern to model

graph. If for any relationship, more than one instances exists in model graph corresponding P_{M_i} contains more than one mappings. In that case more than one instances of design pattern ((F_{M_i})) or part of the design pattern may exist in the model graph. If F_{M_i} is empty, design pattern does not exist in the model graph. If F_{M_i} covers each node of the design pattern, design pattern exists in the model graph. Otherwise, design pattern partially exists in the model graph.

4.1 Design Pattern Detection as Strategy Design Pattern: Exact Matching

Firstly, we are considering strategy design pattern, the UML diagram corresponding to it is shown in Fig. 6. There are two relationships direct association relationship and aggregation relationship. Relationship graphs are shown in Fig. 7 and Fig. 8 respectively.

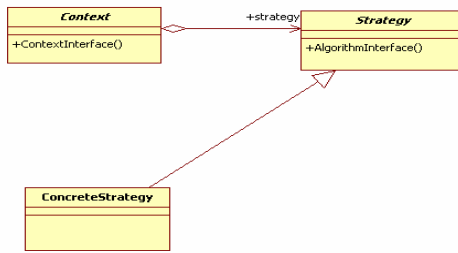


Fig. 6. Strategy Design Pattern

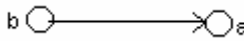


Fig. 7. Generalization Relationship Graph of UML Diagram of Strategy Design Pattern

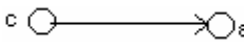


Fig. 8. Aggregation Relationship Graph of UML Diagram of Strategy Design Pattern

Now consider the generalization relationship graph of model graph (MG) shown in Fig. 2 and generalization relationship graph of strategy design pattern (DPG) shown in Fig. 7. Apply the Match_DPG_MG algorithm on these graphs MG and DPG. Initially $Match_DPG_MG(s) = \emptyset$ therefore $Match_DPG_MG(s)$ does not cover all the nodes of DPG, so $T^{out}_1(s) = \emptyset$, $T^{out}_2(s) = \emptyset$, $T^{in}_1(s) = \emptyset$ and $T^{in}_2(s) = \emptyset$. Therefore by equation (3), we have $P(s) = ((1, a), (2, a), (3, a))$.

Now we will calculate $F(s, 1, a)$. Here the nodes connected to '1' in MG are '2' and '3' and the node connected to node 'a' in DPG is 'b'. Since these nodes are not in the current partial matching because $M_1(s) = \emptyset$ and $M_2(s) = \emptyset$ the algorithm does not check each branch from or to '1' has a corresponding branch from or to 'a' and vice versa. So the algorithm goes to otherwise condition, the number of nodes in $T^{in}_1(s)$, $T^{out}_1(s)$, $T^{in}_2(s)$, $T^{out}_2(s)$ are zero and the number of nodes in $(N_1 - M_1(s))$

- $T^{in}_1(s) - T^{out}_1(s)$ is 3, the number of nodes in $(N_2 - M_2(s) - T^{in}_2(s) - T^{out}_2(s))$ is 2. Since the number of nodes in N_1 are more than N_2 . Thus $F(s, 1, a) = \text{true}$. Therefore, there is graph-sub graph isomorphism. We get $s' = (1, a)$ thus $P_M = (1, a)$. Now Call $\text{Match_DPG_MG}(s')$. Since $(1, a)$ does not cover all the nodes of DPG, we have to again compute the set $P(s)$ of the pairs candidate. To compute $P(s)$ we have to find Pred and Succ of the nodes in MG and DPG respectively. $\text{Pred}(G1, 1) = (2, 3)$, $\text{Succ}(G1, 1) = (\emptyset)$, $\text{Pred}(G2, a) = (b)$, $\text{Succ}(G2, a) = (\emptyset)$. Now we will find $T^{out}_1(s)$, $T^{in}_1(s)$, $T^{out}_2(s)$, $T^{in}_2(s)$. $T^{out}_1(s) = (\emptyset)$, $T^{in}_1(s) = (2, 3)$, $T^{out}_2(s) = (\emptyset)$, $T^{in}_2(s) = (b)$. Here both $T^{out}_1(s)$ and $T^{out}_2(s)$ are empty and both $T^{in}_1(s)$ and $T^{in}_2(s)$ are not empty. Therefore, the equation (2) is satisfied. So $P(s) = ((2, b), (3, b))$.

Tracing further the $\text{Match_DPG_MG}(s)$ algorithm, we find that $P_M1 = \{((1, a), (2, b)), ((1, a), (3, b))\}$ for the generalization relationship. Now we will consider the next loop iteration for $(2, a) \in P(s)$, calculate $F(s, 2, a)$ in the similar way as we discussed above for $F(s, 1, a)$. Here the node connected to 2 in MG is 1 and a in DPG is b. These nodes are in the current partial matching because $1 \in M_1(s)$ and $b \in M_2(s)$ but there is no branch from 2 to 1 has a corresponding branch from a to b. Therefore, the Boolean function $F(s, 2, a)$ is not true. Hence the loop is terminated and then check for another condition $(3, a) \in P(s)$. Similarly for $(3, a)$ the Boolean function is not true. Proceeding in the same manner we could find two instances $P_M1 = \{((1, a), (2, b)), ((1, a), (3, b))\}$ of the generalization relationship graph of the strategy design pattern in the model graph.

Now consider the aggregation relationship of model graph (MG) shown in Fig. 4 and aggregation relationship of strategy design pattern (DPG) shown in Fig. 8. Proceeding in same way as above we could find that $P(s) = ((1, a), (5, a))$ and $P_M2 = \{(1, a), (5, c)\}$. Now we will scan the loop iteration for $(5, a) \in P(s)$. $T^{out}_1(s) = 1$, $T^{in}_1(s) = (\emptyset)$, $T^{out}_2(s) = (\emptyset)$, $T^{in}_2(s) = (c)$. Since here none of the condition is satisfied. So the further exploration is avoided and there is no match found [10]. Thus only one instance $P_M2 = \{(1, a), (5, c)\}$ of the Aggregation relationship graphs of the Strategy design pattern exist in the model graph. Here both of the relationship graphs of the Strategy design pattern exist in the model graph. For detecting the whole pattern, now apply the $\text{Union_Relationships}$ algorithm on $P_M1 = ((1, a), (2, b))$ of generalization relationship and $P_M2 = ((1, a), (5, c))$ of the aggregation relationship, the pair $(1, a)$ of generalization relationship matches with the aggregation relationship and also all the nodes have the corresponding edges. So $F_M1 = ((1, a), (2, b), (5, c))$. Similarly for next pair $P_M1 = ((1, a), (3, b))$ and $P_M2 = ((1, a), (5, c))$, $F_M2 = ((1, a), (3, b), (5, c))$. Thus, two instances of the Strategy Design pattern completely exist in the model graph.

4.2 Design Pattern Detection as Command Design Pattern: Partial Matching

In some cases it is also possible that a particular design pattern partially exists in the system design pattern (case ii discussed in section 4). For example consider the command design pattern, the UML diagram corresponding to it, is shown in Fig. 9.

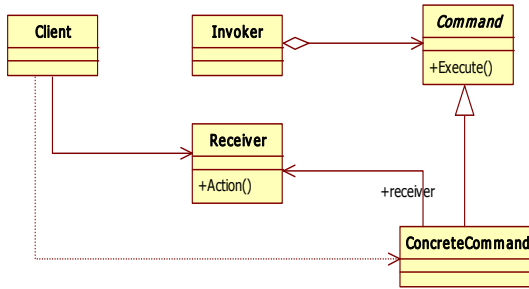


Fig. 9. Command Design Pattern

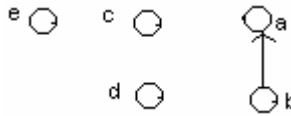


Fig. 10. Generalization Relationship Graph of UML Diagram of Command Design Pattern

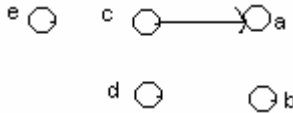


Fig. 11. Aggregation Relationship Graph of UML Diagram of Command Design Pattern

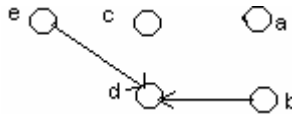


Fig. 12. Direct Association Relationship Graph of UML Diagram of Command Design Pattern

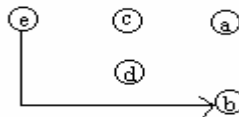


Fig. 13. Dependency Relationship Graph of UML Diagram of Command Design Pattern

There are four relationships, i.e. generalization relationship, direct association relationship, aggregation relationship, and dependency relationship, relationship graph has been shown in Fig.10, Fig. 11, Fig. 12, and Fig. 13 respectively.

For the generalization, aggregation and dependency relationship graphs of the model graph and Command design pattern if we trace the Match_DPG_MG(s) algorithm, we find that $P_M_1 = \{(1, a), (2, b)\}, \{(1, a), (3, b)\}$ for the generalization,

$P_{M_2} = \{(1, a), (5, c)\}$ for the aggregation, and $P_{M_3} = \{(6, b), (3, e)\}$ for the dependency.

Now consider the direct association relationship graph of model graph (MG) shown in Fig. 3 and direct association relationship graph of command design pattern (DPG) shown in Fig. 12, and apply the $Match_DPG_MG(s)$ algorithm on these MG and DPG graphs. We find that $P(s) = \{(2, b), (3, b), (4, b), (5, b)\}$ and for $(2, b) \in P(s)$, we get $T^{out}_1(s) = (\emptyset)$, $T^{in}_1(s) = 4$, $T^{out}_2(s) = d$, $T^{in}_2(s) = (\emptyset)$. Since none of the condition is satisfied, the further exploration is avoided and there is no match found [10]. Similarly we iterate for $(3, b), (4, b)$ and $(5, b) \in P(s)$ and we find that no condition is satisfied for any of these pairs, so the further exploration is avoided and no match is found [10]. Therefore it can be said that the direct association relationship graph of the command design pattern does not exist in the corresponding direct association relationship graph of the model graph. Now apply the Union_Relationships algorithm. We can find that $F_M = \{F_{M_1}, F_{M_2}\}$, where $F_{M_1} = ((1, a), (2, b), (5, c))$ and $F_{M_2} = ((1, a), (3, b), (5, c))$.

Thus, we find out that out of four relationship graphs of the command design pattern, three relationship graphs (i.e., generalization relationship graph, aggregation relationship graph and dependency) exist in the corresponding relationship graphs of the model graph, but the direct association relationship graph of the command design pattern graph does not exist in the corresponding direct association relationship graph of the model graph. This is the case of partial matching of design pattern in the model graph.

4.3 Particular Design Pattern May or May Not Exist

Above we have seen the examples of design pattern existence but it can be possible that a particular design pattern does not exist in the model graph. In this case we will not find any partial matching solution which covers all the nodes of the design pattern graph. For example if we take singleton design pattern (Fig. 14). There is only one relationship: direct association relationship on itself node. This is shown in following Fig. 15.

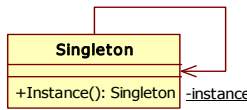


Fig. 14. Singleton Design Pattern



Fig. 15. Direct Association Relationship Graph of UML Diagram of Singleton Design Pattern

Now consider the direct association relationship graph of UML diagram of the model graph i.e., Fig. 3 and direct association relationship graph of UML diagram of the singleton design pattern i.e., Fig. 14. Let graph of Fig. 3 is MG and graph of Fig. 14 is DPG and apply the graph-matching algorithm for both of the graphs MG and DPG. On applying the $\text{Match_DPG_MG}(s)$ algorithm, we find that $P(s) = \{(2, a), (3, a), (4, a), (5, a)\}$, and $P_M_1 = \{(2, a), (3, a), (4, a), (5, a)\}$. On applying the $\text{Union_Relationships}$ algorithm, for all pairs of direct association relationship of design pattern and model graph, there is no corresponding edges are found. Therefore, singleton design pattern does not exist in the model graph.

5 Related Work

The first attempt for automatically detecting design pattern was by Brown [7]. In this work, Smalltalk code was reverse-engineered to facilitate the detection of four well-known patterns from the catalog by Gamma et al. [1]. Antoniol et al. [9] developed a technique to identify structural patterns in a system to observe how useful a design pattern recovery tool could be in program understanding and maintenance. Nikolaos Tsantalis [2] proposed a methodology for design pattern detection using similarity scoring. However, the limitation of similarity algorithm is that it only calculates the similarity between two vertices, not the similarity between two graphs.

Jing Dong [3] gave another approach called template matching, which calculates the similarity between sub graphs of two graphs instead of vertices, to solve the above limitation. S. Wenzel [4] purposed a difference calculation method works on UML models. The advantage of difference calculation method on other design pattern detecting technique is that it detects the incomplete pattern instances also. Bergenti and Poggi [10] developed a method that examines UML diagrams and proposes the software architect modifications to the design that lead to design patterns. In our earlier work [11-15], we have shown how to detect design patterns using various techniques. The space complexities of previous algorithms were very high. We are trying to reduce this complexity by using the graph-matching algorithm [5].

6 Conclusions

A new approach for design pattern detection using state space representation of graph matching has been proposed. In this paper we took the relationship graphs of the model graph (MG) and a design pattern (DPG), after that the graph matching algorithm $\text{Match_DPG_MG}(s)$ is recursively applied on both of the graphs for each relationships and tried to find out the partial matching solution. Then partial mappings (P_M_i) of each relationship graphs are combined to get final mapping (F_M) using $\text{Union_Relationships}$ algorithm. If this final mapping covers all the nodes of the design pattern graph, we say that the design pattern exist in the model graph. If the final mapping (F_M) is empty, design pattern does not exist in the model graph. Otherwise, design pattern partially exists in the model graph. If more than one final mapping (F_M_i) exists, more than one instances of the design pattern or its part exist in model graph. The advantage of this approach is that its memory requirement is only $O(N)$, with a small constant factor.

References

1. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: *Design Patterns Elements of Reusable Object-Oriented Software*. Addison- Wesley, Reading (1995)
2. Tsantalis, N., Chatzigeorgiou, A., Stephanides, G., Halkidis, S.: Design Pattern Detection Using Similarity Scoring. *IEEE Transaction on software Engineering* 32(11) (2006)
3. Dong, J., Sun, Y., Zhao, Y.: Design Pattern Detection by Template Matching. In: *The Proceedings of The 23rd Annual ACM Symposium on Applied Computing (SAC)*, Ceara, Brazil, pp. 765–769 (2008)
4. Wenzel, S., Kelter, U.: Model-driven design pattern Detection using difference calculation. In: *Proc. of the 1st International Workshop on Pattern Detection for Reverse Engineering (DPD4RE)*, Benevento, Italy (2006)
5. Cordella, L.P., Foggia, P., Sansone, C., Vento, M.: An Improved Algorithm for Matching Large Graphs, Dipartimento di Informatica e Sistemistica Universita degli Studi di Napoli “Federico II” Via Claudio, 21 – 80125 Napoli ITALY
6. Nilsson, N.J.: *Principles of Artificial Intelligence*. Springer, Heidelberg (1982)
7. Brown, K.: *Design Reverse-Engineering and Automated Design Pattern in Smalltalk*. Technical Report TR-96-07, Dept. of Computer Science, North Carolina State Univ. (1996)
8. Antoniol, G., Casazza, G., Di Penta, M., Fiutem, R.: Object-Oriented Design Patterns Recovery. *J. Systems and Software* 59(2), 181–196 (2001)
9. Cordella, L.P., Foggia, P., Sansone, C., Vento, M.: Performance evaluation of the VF Graph Matching Algorithm. In: *Proc. of the 10th ICIAP*, pp. 1172–1177. IEEE Computer Society Press, Los Alamitos (1999)
10. Bergenti, F., Poggi, A.: Improving UML Designs Using Automatic Design Pattern Detection. In: *Proc. 12th Int’l Conf. Software Eng. and Knowledge Eng. SEKE 2000* (2000)
11. Pande, A., Gupta, M.: Design Pattern Detection Using Graph Matching. *International Journal of Computer Engineering and Information Technology (IJCEIT)*, 15(20), Special Edition, 59–64 (2010)
12. Pande, A., Gupta, M.: Design Pattern Mining for GIS Application using Graph Matching Techniques. In: *3rd IEEE International Conference on Computer Science and Information Technology, Chengdu, China*, pp. 9–11 (2010)
13. Pande, A., Gupta, M., Tripathi, A.K.: A New Approach for Detecting Design Patterns by Graph Decomposition and Graph Isomorphism. In: *International Conference on Contemporary Computing*, Jaypee Noida, CCIS, Springer, Heidelberg (2010)
14. Pande, A., Gupta, M., Tripathi, A.K.: A Decision Tree Approach for Design Patterns Detection by Subgraph Isomorphism. In: Das, V.V., Vijaykumar, R. (eds.) *ICT 2010. Communications in Computer and Information Science*, vol. 101, Springer, Heidelberg (2010)
15. Pande, A., Gupta, M., Tripathi, A.K.: DNIT – A New Approach for Design Pattern Detection. In: *International Conference on Computer and Communication Technology (ICCCT-2010)*, proceedings to be published by the IEEE (accepted 2010)

MST-Based Cluster Initialization for K -Means

Damodar Reddy, Devender Mishra, and Prasanta K. Jana*

Department of Computer Science & Engineering
Indian School of Mines, Dhanbad 826 004, India
damumtech@gmail.com, devendermishra@ismu.ac.in,
prasanta@isima.org

Abstract. Clustering is an exploratory data analysis tool that has gained enormous attention in the recent years specifically for gene expression data analysis. The K -means clustering is a method of cluster analysis which aims to partition n data points into K clusters. The K -means is possibly the best known and most widely used clustering technique. However, K -means does not necessarily find the optimal cluster configuration due to its significant sensitiveness in random selection of the initial cluster centers. On the other hand, MST-based clustering algorithm suffers from the selection of the inconsistent edges to produce quality clusters. In this paper, we present a novel method that bridges the K -means and the MST-based clustering algorithms. The proposed method not only overcomes the problem of random selection of the initial cluster centers for the former and the inconsistent edges for the later one but also automate them. We perform extensive experiments on the proposed method using both synthetic as well as the real world data sets. The experimental results show that the algorithm is able to produce desired clusters even for complex and high dimensional data points.

Keywords: Clustering, K -means algorithm, Minimum cost spanning tree, Validity index.

1 Introduction

Clustering is a process of organizing objects into meaningful groups called clusters such that the objects within the groups are similar in nature where as those in different groups are dissimilar. Clustering is a powerful tool that has been applied in various fields including image processing, machine learning, wireless sensor networks, medicine and economics [1], [2], [3]. Data modeling puts clustering in a historical perspective rooted in mathematics, statistics, and numerical analysis. There has been a tremendous growth of several biological data. Identifying similar data to form various meaningful groups is extremely important in the field of computational biology. Specifically, clustering is an essential tool in microarray or gene expression data analysis. Therefore clustering analysis for biological data has become an essential and valuable tool in the field of computational biology / bioinformatics [4]. A rich literature has been developed over the past decades on clustering analysis, a survey of which can be

* IEEE Senior Member.

found in [5], [6], [7]. However, most of these algorithms do not work for a number of large data sets with large dimensions. Moreover, because of the huge variety of the data distribution, no techniques are completely satisfactory and most of them are confronted in meeting the robustness, quality and fastness at the same time. As a result, researchers have paid huge attention to design more effective clustering algorithms.

1.1 K-Means Algorithm

K -means [8] is possibly the most popular technique among the partitional clustering algorithms. The algorithm partition n data points $\{x_1, x_2, \dots, x_n\}$ into K clusters. The pseudo code of K -means algorithm consists of a simple re-estimation procedure as follows [9]:

Step 1: Select randomly K initial cluster centers c_1, c_2, \dots, c_K from the given n points $\{x_1, x_2, \dots, x_n\}$, $K \leq n$.

Step 2: Assign each point x_i , $i = 1, 2, \dots, n$ to the cluster C_j corresponding to center c_j , for $j = 1, 2, \dots, n$ iff $\|x_i - c_j\| \leq \|x_i - c_p\|$ for $p = 1, 2, \dots, K$ and $j \neq p$.

Step 3: Compute new cluster centers $c_1^*, c_2^*, \dots, c_K^*$ as follows

$$c_i^* = \frac{1}{n_i} \sum_{x_j \in C_i} x_j \quad \text{for } i = 1, 2, \dots, K$$

where n_i is the number of elements belonging to cluster C_i .

Step 4: If $c_i^* = c_i, \forall i = 1, 2, \dots, K$ then terminate; otherwise, continue from step 2.

Although the K -means has been widely accepted due its simplicity and robustness, it generally suffers from two main drawbacks. 1) It requires the user to input the number of clusters which is usually difficult to know in advance for many real world data sets. 2) The results of K -means algorithm depends on the cluster initialization which is random.

1.2 MST-Based Algorithm

Minimum spanning tree (MST) is a useful graph for detecting clusters of a given set of data points. MST has been well suited for clustering in the fields of image processing [10], pattern recognition [11], and computational biology [12]. Cluster design using MST was initiated by Zahn [13]. Removal of some edges from the MST result into some connected components that represent clusters as shown in Fig. 1(a) and Fig. 1(b). By definition, each edge in the MST of a set of points is the smallest edge connecting two points. Thus points in different clusters should intuitively be connected by longer edges in the MST than points in the same cluster. So, cutting the largest edges in the MST may separate the most prominent clusters of points. When the heaviest edges in an MST are cut, we obtain homogeneous clusters. Cutting by edge length alone may not be sufficient to split the correct clusters, because the clusters thus obtained may not be of reasonable size and/or concentration. To obtain good clusters, one needs to impose some additional constraints on the edges to be cut. Usually, the MST-based clustering algorithms follow the three basic steps [13]: 1) A minimum spanning tree is constructed on the given set, 2) the inconsistent edges are

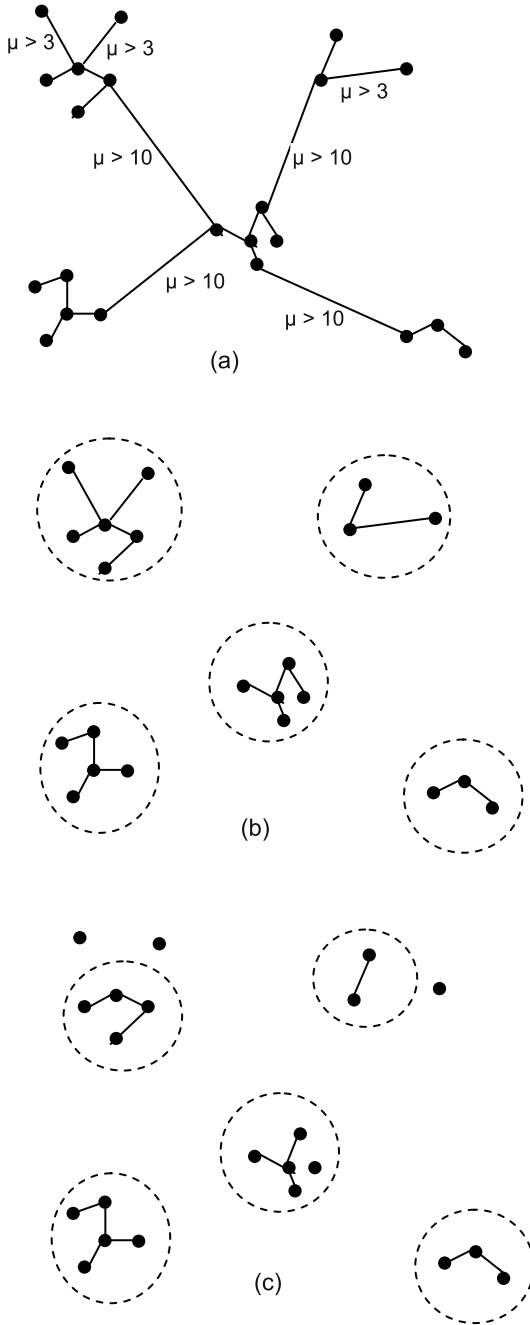


Fig. 1. (a) Original Minimum spanning tree, all edges are not labeled (b) Cluster formation as shown by dashed circles after removal of edges of length $\mu > 10$ (c) Some points are isolated from the clusters when the edges are removed for $\mu > 3$

selected and removed to obtain a set of connected components (clusters) and 3) Step 2 is repeated until some termination condition is satisfied. The main advantage of MST-based clustering algorithm is that it is capable of detecting irregular boundaries [14] and unlike other clustering algorithms, it does not assume any spherical structure of the underlying data [15]. However, the significant demerit goes to this technique is that the improper selection of the inconsistent edges may isolate some of the points from a cluster. This is illustrated in Fig. 1.

1.3 Objective

In this paper, we propose a cluster initialization method that overcomes the problems of random initialization of clusters for K -means and removal of the inconsistent edges for MST-based algorithm. Our approach makes a link between the K -means and MST-based clustering algorithm as follows. Given the n data points, we first construct the minimum spanning tree and try to generate some initial clusters (connected components) by removing some edges with some threshold value. We find some initial cluster centers from these clusters and then go on refining them by successive runs of K -means algorithm. The refining process is performed by incorporating the validity index (defined later) inside the runs of the K -means and the MST-based algorithm.

1.4 Validity Index

Validity index is generally used to measure the quality of the clusters. The validity Index used in our algorithm is based on the compactness and isolation of the clusters. Compactness of the clusters is measured by the Intra-cluster distance whereas isolation between the clusters is by Inter-cluster distance defined as follows.

Intra-cluster distance: It is the average distance of all the data points within a cluster from the cluster center and is calculated by

$$intra_dist = \frac{1}{N} \sum_{i=1}^K \sum_{x \in C_i} \|x - c_i\|^2 \quad (1.1)$$

where c_i is the center of the cluster C_i and N is the number of data points.

Inter-cluster distance: It is the minimum of the pair wise distance between any two cluster centers and is calculated by

$$inter_dist = \text{Min} \|c_i - c_j\|^2 \quad (1.2)$$

where $i = 1, 2, \dots, K - 1$; $j = i + 1, i + 2, i + 3, \dots, K$;

Using the above distances in equation 1.1 and 1.2, we use the following validity index as proposed by Ray and Turi [34]:

$$Val_index = \frac{intra_dist}{inter_dist} \quad (1.3)$$

The rest of the paper is organized as follows. A review of the K -means along with its extensions is discussed in section 2. The proposed algorithm is presented in section 3. The experimental results are described in section 4 followed by the conclusion in section 5.

2 Related Work

An enormous research has been carried out over the extension of the K -means, the review of which can be found in [16]. Various attempts have also been made towards the cluster initialization of K -means algorithm. One of the oldest works is due to Forgy [17]. Tou and Gonzales [18] reported the Simple Cluster-Seeking (SCS) method that chooses the seed points iteratively using some threshold value for the distance between two seeds. Binary Splitting (BS) Method developed by Linde et al. [19] generates the initial cluster centers with the successive runs of K -means algorithm itself. Babu and Murty [20] developed a method of near optimal seed selection using genetic programming. Pelleg and Moore [21] proposed an algorithm based on kd-tree that efficiently identifies the closest cluster centers for all the data points. Huang and Harris [22] proposed Direct Search Binary Splitting (DSBS) which is based on Binary Splitting Algorithm. This method chose the ϵ vector of Linde et al. [19] and is very useful to improve the cluster quality. Katsavounidis et al. [23] developed an algorithm called KKZ algorithm that chose K seeds based on edges of the data. Daoud and Roberts [24] proposed a method that divides the given data points into M disjoint subspaces and calculate seeds for each sub space. Thiesson et.al [25] suggested a method for mean based production of randomized seeds. Bradley and Fayyad [26] proposed an initialization method that randomly divides the given data set into certain groups and K -means is applied with same set of initial seeds that are taken from Forgy's method. Likas et al. [27] developed global K -means algorithm based on seeds increment and kd-tree. Khan and Ahmed [28] developed a method called Cluster Center Initialization Method (CCIA) that is mainly based on Density-based Multi Scale Data Condensation (DBMSDC) proposed by Mitra et al. [29]. All the above initialization methods are well gathered by Stephen J. Redmond, Conor Heneghan [30]. There are also some other interesting algorithms [31], [32], [33] that describe the initialization method for K -means algorithm.

3 Proposed Algorithm

The idea of our proposed algorithm is as follows. In step 1, we construct the minimum spanning tree on the given data sets using Kruskal algorithm. In step 2, we then store the weights (i.e., Euclidean distance) of all the edges in an array called `edge_weight` without any redundant entry. We then assign the first element of the `edge_weight` to μ in step 3. Step 4 removes all the edges of the MST whose weights are greater than μ . Suppose it results into p number of clusters. In step 5, we find the centers of all these clusters and pass them as the new data points to the K -means algorithm. We then run the K -means inputting the number of output clusters varying from 2 to p and store the `Val_index` corresponding to each run. Then we record the least of all these *validity indices* to store it in an array called `Min_V`. In step 6, we set μ to the next weight

from `edge_weight` array and repeat from step 4 unless and until the array `edge_weight` is exhausted. We obtain the final clusters corresponding to the minimum value of the `Min_V` array elements.

We now formalize our algorithm stepwise as follows:

- Step 1: Given n data points, construct the MST using Kruskal algorithm.
- Step 2: Store all the distinct edge weights (Euclian distance) in the array `edge_weight`.
- Step 3: Assign first element of the array `edge_weight` to μ and delete the element from the array.
- Step 4: Find the clusters by removing all the edges of the MST (constructed in step 1) whose length is more than μ . Assume the number of clusters formed in this step is p .
- Step 5: Find the p centers of the clusters formed in step 4.
- Step 6: Pass the p centers found in step 5 as the new data points to K -means algorithm varying the number of output clusters from 2 to p and obtain the `Val_index` for each run of the K -means algorithm and store the minimum one in the array `Min_V`.
- Step 7: Assign the next element of the array `edge_weight` to μ and delete the element from the array.
- Step 8: If the array `edge_weight` is not exhausted then go to Step 4.
- Step 9: Find the minimum value of the array `MIN_V` and obtain the final clusters corresponding to this value.
- Step 10: stop.

This can be noted that like the Linde's algorithm [19] our algorithm also has the increased computational complexity due to the repeated run of the K -means algorithm. However, this can be further reduced in several ways. For example, we can fix a threshold value (say, τ) for validity index and use it to impose some termination criterion say, $\text{val_index} \leq \tau$ in step 6. Our future research will be carried out to improve the computational complexity opting such strategic methods.

4 Experimental Results

In order to test our algorithm, we consider several synthetic as well as real world data sets. For the sake of visualization, we first take random data of 1890 points and apply our proposed algorithm and K -means as well. The experimental results are shown in Fig. 2(a) and Fig. 2(b) respectively. Both the algorithms successfully created five well separated clusters as shown in five different colors with same color inside a cluster. Likewise we apply our algorithm and the K -means on some other data sets namely, kernel, banana and 2-spiral which are known to be complex for clustering. The experimental results are shown in Figs. 3, 4 and 5 respectively. In case of kernel data of

size 550 points, our algorithm results two well separated clusters as shown in Fig.3 (a), however, K -means fails to obtain the desired clusters as it produced same number of clusters with half of the data points taken from each of the clusters as in Fig. 3(b). Similarly, in case of banana data, our algorithm forms two clusters (Fig. 4(a)) as the data set looks like two bananas. In this case K -means created two clusters with overlapping as shown in Figs. 4(b). Finally in 2-spiral data set the data points are in the form of two spiral shaped curves (clusters). Clearly our method results into two clusters as shown in Fig. 5(a) where as the K -means produces the same number of clusters but with overlapping manner as shown in Fig. 5 (b) which is not desirable.

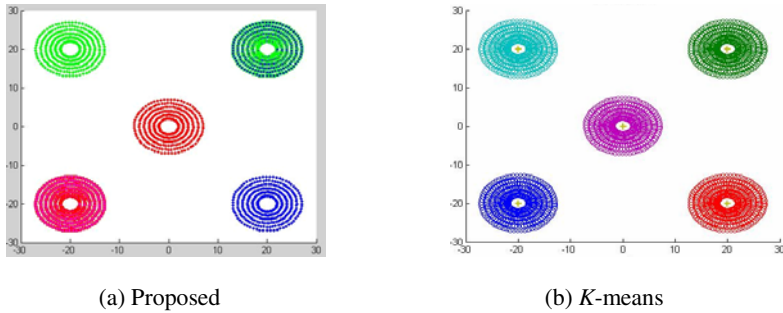


Fig. 2. Results on synthetic data of size 1890 points

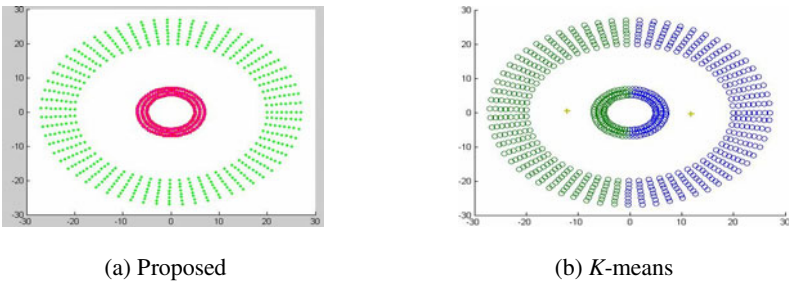


Fig. 3. Results on kernel data of size 550 points

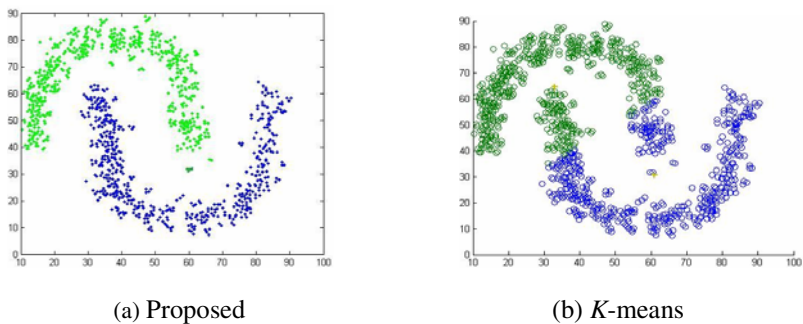


Fig. 4. Results on banana data of size 1000 points

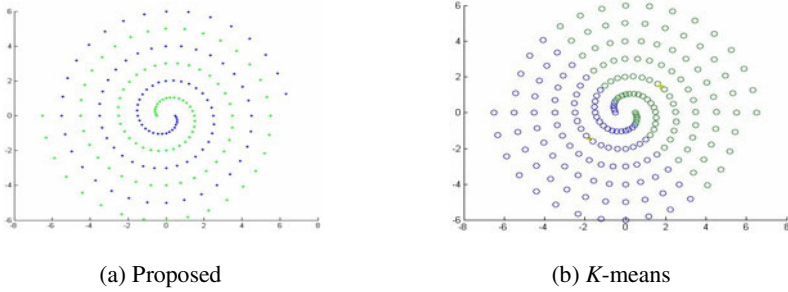


Fig. 5. Results on 2-spiral data of size 200 points

We also run our algorithm for a few real world data sets which are multi-dimensional. They are Irish, Wine, Spect Heart and Ecoli from UCI machine learning repository [35]. The descriptions of such datasets can be found in [35]. The comparison between the *K*-means and our proposed method is shown in Table 1. It is obvious to note that our proposed algorithm is better than the the *K*-means algorithm as evident by the validity.

Table 1. Comparison of the proposed method with the *K*-means on the real world data sets with respect to validity index

Name	Data size	No.of clusters	Val_Index (<i>K</i> -means)	Val_Index (Ours)
Iris	150	3	0.2888	0.1562
S.Heart	187	2	1.8806	0.5101
Wine	178	3	0.1895	0.1153
Ecoli	336	8	1.3412	0.9835

5 Conclusions

In this paper, we have presented a new method for the initial cluster selection which leads to the best results from *K*-means algorithm. The method uses the MST-based clustering and the *K*-means. We have performed excessive experiments on the artificial as well as the real world data sets. The results in terms of validity index are very encouraging and can compete with few advanced algorithms. However, the method has increased computational complexity. Our future attempt will be made to improve this complexity.

Acknowledgments. This work has been supported by the Council of Scientific and Industrial Research (CSIR) under the grant (No. 25(0177) / 09 / EMR-II). The authors would like to thank CSIR for the support.

References

1. Abbasi, A.A., Younis, M.: A Survey on Clustering Algorithms for Wireless Sensor Networks. *Computer Communications*, 2826–2841 (2007)
2. Villmann, T., Albani, C.: *Clustering of Categorical Data in Medicine—Application of Evolutionary Algorithms*. Springer, Heidelberg (2006)
3. Garibaldi, U., Costantini, D., Donadio, S., Viarengo, P.: Herding and Clustering in Economics: The Yule-Zipf-Simon Model. In: *Computational Economics*, vol. 27, pp. 115–134. Springer, Heidelberg (2006)
4. Madeira, S., Oliveira, A.L.: Biclustering Algorithms for Biological Data Analysis: A Survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 1, 24–45 (2004)
5. Kerr, G., Ruskina, H.J., Crane, M., Doolan, P.: Techniques for Clustering Gene Expression Data. *Computers in Biology and Medicine* 38, 283–293 (2008)
6. Mitra, S., Banka, H.: Multi-objective Evolutionary Biclustering of Gene Expression Data. *Pattern Recognition* 39, 2464–2477 (2006)
7. Jiang, D., Tang, C., Zhang, A.: Cluster Analysis for Gene Expression Data. *IEEE Transactions on Knowledge and Data Engineering* 16, 1370–1386 (2004)
8. Jain, A.K.: *Algorithms for Clustering Data*. Prentice Hall, Englewood Cliffs (1988)
9. Bandyopadhyay, S., Maulik, U.: An Evolutionary Technique Based on K -Means Algorithm for Optimal Clustering in \mathfrak{R}^N . *Information Science—Applications* 146, 221–237 (2002)
10. Victor, S.P., John Peter, S.: A Novel Minimum Spanning Tree Based Clustering Algorithm for Image Mining. *European Journal of Scientific Research* 40, 540–546 (2010)
11. Zhong, C., Miao, D., Wang, R.: A Graph-theoretical Clustering Method Based on Two Rounds of Minimum Spanning Trees. *Pattern Recognition* 43, 752–766 (2010)
12. Han, A., Zhu, D.: DNA Computing Model for the Minimum Spanning Tree Problem. In: *Eighth International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC 2006)*. IEEE Computer Society, Los Alamitos (2006)
13. Zahn, C.T.: Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters. *IEEE Trans. on Computers* 20(1), 68–86 (1971)
14. He, Y., Chen, L.: MinClue: A MST-based Clustering Method with Auto-Threshold-Detection. In: *Proceedings of the IEEE Conference on Cybernetics and Intelligent Systems*, Singapore (2004)
15. Wang, X., Xiali, W., Wilkes, D.M.: A Divide and Conquer Approach for Minimum Spanning Tree-based Clustering. *IEEE Transactions on Knowledge and Data Engineering* 21, 945–988 (2009)
16. Jain, A.K.: Data Clustering: 50 years beyond K -means. *Pattern Recognition Letters* 31, 651–666 (2010)
17. Forgy, E.W.: Cluster Analysis of Multivariate Data: Efficiency vs. Interpretability of Classifications. *Biometrics* 21, 768–769 (1965)
18. Tou, J.T., Gonzales, R.C.: *Pattern Recognition Principles*. Addison-Wesley, Reading (1974)

19. Linde, Y., Buzo, A., Gray, R.M.: An Algorithm for Vector Quantizer Design. *IEEE Transactions on Communication* 28, 84–95 (1980)
20. Babu, G.P., Murty, M.N.: A Near-optimal Initial Seed Value Selection in K-means Algorithm using a Genetic Algorithm. *Pattern Recognition* 14(10), 763–769 (1993)
21. Pelleg, D., Moore, A.: Accelerating Exact K-means Algorithms with Geometric Reasoning. In: *Proceedings of the fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 277–281. ACM, New York (1999)
22. Huang, C.M., Harris, R.W.: A Comparison of Several Codebook Generation Approaches. *IEEE Transactions on Image Processing*, 108–112 (1993)
23. Katsavounidis, I., Kuo, C.C.J., Zhang, Z.: A New Initialization Technique for Generalized Lloyd Iteration. *IEEE Signal Process Lett.* 1(10), 144–146 (1994)
24. Al-Daoud, M.B., Roberts, S.A.: New Methods for the Initialization of Clusters. *Pattern Recognition Letters* 7, 451–455 (1996)
25. Thiesson, B., Meck, B., Chickering, C., Heckerman, D.: *Learning Mixtures of Bayesian Networks*. Microsoft Technical Report, Redmond, WA
26. Bradley, P.S., Fayyad, U.M.: Refining Initial Points for *K*-means Clustering. In: *Proc. 15th Internat. Conf. on Machine Learning*, San Francisco, CA, pp. 91–99 (1998)
27. Likas, A., Vlassis, N., Verbeek, J.J.: The Global *K*-means Clustering Algorithm. *Pattern Recognition* 36, 451–461 (2003)
28. Khan, S.S., Ahmad, A.: Cluster Center Initialization Algorithm for *K*-means Clustering. *Pattern Recognition Lett.*, 1293–1302 (2004)
29. Mitra, P., Murthy, C.A., Pal, S.K.: Density-based Multiscale Data Condensation. In: *IEEE Transactions in Pattern Analysis and Machine Intelligence*, vol. 24, pp. 734–747. IEEE Computer Society, Los Alamitos (2002)
30. Redmond, S.J., Heneghan, C.: A Method for Initializing the *K*-means Clustering Algorithm using kd-trees. *Pattern Recognition Letters* 28, 965–973 (2007)
31. Pena, J.M., Lozano, J.A., Larranaga, P.: An Empirical Comparison of Four Initialization Methods for the *K*-Means Algorithm. *Pattern Recognition* 20, 1027–1040 (1999)
32. Su, T., Jennifer, D.: A Deterministic Method for Initializing *K*-means Clustering. In: *IEEE International Conference on Tools with Artificial Intelligence* (2004)
33. Lu, J.F., Tang, J.B., Tang, J.M., Yang, J.Y.: Hierarchical Initialization Approach for *K*-Means Clustering. *Pattern Recognition* 25, 787–795 (2008)
34. Ray, S., Turi, R.H.: Determination of Number of Clusters in *K*-means Clustering and Application in Colour Image Segmentation. In: *Proceedings of 4th International Conference (ICAPRDT 1999)*, Calcutta, pp. 137–143 (1999)
35. UCI Machine Learning Repository,
<http://archive.ics.uci.edu/ml/dataset>

Geometry and Skin Color Based Hybrid Approach for Face Tracking in Colour Environment

Mahesh Goyani¹, Gitam Shikkenawis², and Brijesh Joshi³

¹ SP University, Gujarat, India

² DA-IICT, Gandhinagar, Gujarat, India

³ IIT Hyderabad, India

mgoyani@gmail.com, gitam365@yahoo.co.in,

brijesh_joshi001@yahoo.com

Abstract. Face detection and face tracking has been a fascinating problem for image processing researchers during the last decade because of many important applications such as video face recognition at airports and security checkpoints, digital image archiving, etc. In this paper, we attempt to detect faces in a digital image using skin colour segmentation, morphological processing and acceptance/rejection based on face geometry. Face colour segmentation is done with different colour models like RGB and HSV for better results. The same algorithm is later on used for face tracking. Face tracking and video surveillance are some of the noticeable applications of the face detection. Face tracking in offline video is implemented using Skin Colour Segmentation algorithm. We have tested our system for standard face datasets CVL and LFW for face detection. Offline videos are recorded in current working environment with dynamic conditions. Results talk about the robustness of proposed algorithm.

Keywords: Face Detection, Face Recognition, Skin Colour Segmentation, Template Matching, Morphological Operation.

1 Introduction

Many applications in Human Computer Interaction consider the human face as basic entity for authentication [1]. Due to the high inter-personal variability (e.g. gender and race), the intra-personal changes (e.g. pose and expression), and the acquisition conditions (e.g. lighting and image resolution) face detection is not a trivial task and it is still an open problem [2]. Skin colour works as very good information for the face detection. The aim of this paper is to detect the presence or absence of faces in their field of vision.

Many techniques have been found for face detection from the images like knowledge based method (Yang and Huang, 1994), feature based approach (Yang and Waibel, 1996; Kjeldsem and Kinder, 1996), template based method (Crawal et. al., 1992; Lanitis et. al., 1995) and appearance based method (Turk and Pentland, 1991; Rowley et. al., 1998; Osuna et. al., 1997) [3].

Our system uses feature based method, which takes skin colour as a feature. Skin colour processing is computationally cheaper compare to other face features [4], [5]. When building the skin colour based system, three basic problems are on the road. Computer, Television like output device uses RGB colour model to display the information. First

problem is the selection of colour space. TSL (*Tint-Saturation-Luma*) model is considered as the best colour space for human eye perception, but RGB to TSL Colour space transformation is very time consuming. So we have used HSV (*Hue-Saturation-Value*) colour space identical to TSL colour space. It takes less time for transformation compare to TSL model. Second problem is the formulation rules for skin colour detection. How exactly the colour distribution should be modeled? And third problem is the processing of region of interest (ROI). How to extract out the face or faces from the segmented skin colour region [6]. Many more skin area like neck, hands, legs or other body part plus objects with skin colour could be the dimension of difficulty in face extraction from ROI [4].

1.1 Database Description

We have used CVL dataset for our experiment, which contains total 11487 images (114 persons with 7 images of each). Another dataset named LFW (Labeled Faces in Wild) is also used. The dataset contains more than 13,000 images of faces collected from the web. 1680 of the people pictured have two or more distinct photos in the data set. For face tracking we have produced our in-house dataset, having 30 videos of 6 subjects with dynamic lighting condition with screen resolution of 640 X 480 with frame rate of 15 per second. Accuracy of our algorithm is more than 90 %.

2 Detection and Classification

As mentioned in [4], we have used skin colour as a face feature. Previous work is extended here for the real time application of face detection. Face detection algorithm is discussed here in short. Algorithmic steps are shown in figure 1. It works in three stages.

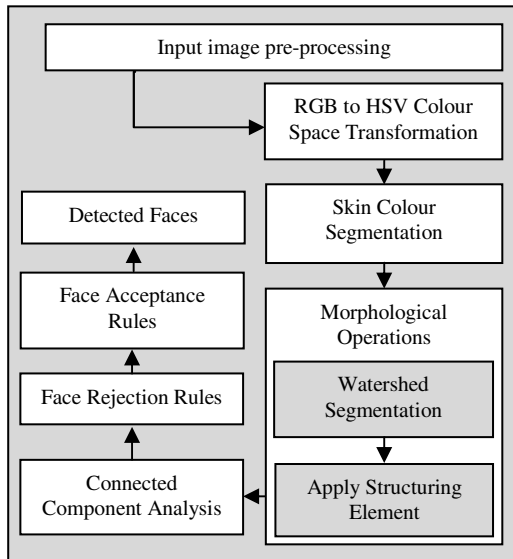


Fig. 1. Face Detection Algorithm (Previous work)

2.1 Skin Colour Segmentation

Skin colour is most commonly used information in face detection in colour images [7], [8], [9]. Colour is very useful cue to extract skin region and it is only available in colour images. It allows face detection without considering much about geometry and texture feature of face [10]. We have chosen skin colour as a processing feature because colour is almost invariant to scale, orientation and partial occlusion [6]. In RGB space, the skin colour region is not well distinguished in all 3 channels. A simple observation of its histogram shows that it is uniformly spread across a large spectrum of values. As shown in fig. 2, H (Hue) and S (Saturation) channel shows significant discrimination of skin colour regions [7]. RGB channels do not provide much information, but HS channel gives good discrimination of skin colour in plane. Colour histogram is global statistical measure of an image [9].

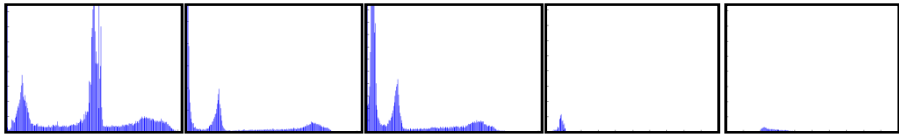


Fig. 2. Skin signature estimation for R, G, B H and S Channel for test image

HSV colour model is most compatible with human perception system [10]. RGB colour space is represented by cube and HSV colour space is defined by cone. So RGB to HSV colour space transformation is non linear. H and S components give chromatic information and V gives illumination information. H and S are independent of V, so with different illumination and same colour, RGB gives different chromaticity value but HSV model exhibit the same value which gives robustness to the HSV colour model selection. The transformation is shown below:

$$V = \max(R, G, B) \tag{1}$$

$$S = \begin{cases} \frac{(V - \min(R, G, B)) \times 255}{V}, & \text{if } V \neq 0 \\ 0, & \text{if } V = 0 \end{cases} \tag{2}$$

$$H = \begin{cases} \frac{(G - B \times 60)}{S}, & \text{if } V = R \\ 180 + \frac{(B - R) \times 60}{S}, & \text{if } V = G \\ 240 + \frac{(R - G) \times 60}{S}, & \text{if } V = B \end{cases} \tag{3}$$

Term H is very significant in environment with red colour component. The reddish portion of image appears, as brightest area after applying H. Skin colour has dominant

red colour property and hence this colour space provides boundary to the skin colour and non-skin colour regions. Figure 3(a) and 3(b) shows RGB colour space image and skin colour segmented image after applying threshold.

$\forall \text{Skin_Pixel (ROI (Test Image))}$,

$$\begin{aligned} BW(i, j) &= 255, \quad \text{if } Ig(\text{Skin_Pixel}) > Tb \\ &= 0, \quad \text{otherwise} \end{aligned} \quad (4)$$

Where, Ig is the intensity of the gray pixel and Tb is the threshold employed for binarization.



Fig. 3. (a). RGB Colour Image, (b). Skin colour segmented image

Generally precise shape based face detection techniques are applied after skin colour segmentation [10].

2.2 Morphological Operations

Morphological operations work with intensity images, so skin colour image needs to be converted in gray scale image. Binary outcome of previous step is used to mask original image, to extract out gray scale image of ROI. Intensity thresholding is performed to break up dark regions into many smaller regions so that they can be cleaned up by morphological opening. The threshold is set low enough so that it doesn't chip away parts of a face but only create holes in it. Morphological opening is performed to remove very small objects from the image while preserving the shape and size of larger objects in the image [11]. The mask image will contain many cavities and holes in the faces. Morphological opening is performed to remove small to medium objects that are safely below the size of a face [12].



Fig. 4. (a). Morphological Opening, (b). After Filling Holes

2.3 Face Acceptance / Rejection Rules

After doing connected component analysis, we are getting many regions on the image as shown in figure 5(a). It includes face and non face regions. We need some rules to reject the non face regions. Generally face height is multiple of its width by some factor, say f . We have set the range of this factor f as it is not always fixed. Factor f is derived by examining fifty images. We are calculating the bounding box around each connected component.

1. Decide rectangle bonding box for all possible connected component
2. Accept it as a face if it satisfies all the following rules.
 - 2.1. Height to Width ration satisfies the range of f .
 - 2.2. If area of bounding box is greater than area threshold Ta .



Fig. 5. (a) Connected Component, (b) Rejection of non faces (c) Output

In figure 5(a), Twenty Five connected components are detected, but only face satisfies the acceptance-repentance rules. Nine components satisfy the rule 2.1 but fail to satisfy the rule 2.2, as they are having smaller area.

3 Proposed Tracking Scheme

Since last decade, Face Detection has been emerged as one of the prominent research area for security. There are so many applications of face detection. We have tried to develop one of the applications of face detection i.e. face tracking using the skin color based algorithm. Face tracking extends face detection to video sequences. Any individual who appears in a video for any length of time generates a face track – that is, a sequence of face instances across time. The goal of face tracking then is to aggregate single-frame detection results into a collection of face tracks. In face tracking we exploit spatio-temporal continuity to associate face instances across frames, and iteratively update motion models for all face tracks, respecting shot boundaries in the process.

For face tracking application, we have employed some set of rules for skin detection. Some of the modifications are proposed in [4]. Here we have considered RGB-HSV values instead of only HSV colour space. This modification enhances the skin region. Two thresholds for each plane (R, H and S) are found for good approximation of ROI. T_{su} and T_{sl} is upper and lower threshold respectively for saturation plane. T_{hu} and T_{hl} define the upper and lower bound respectively for the Hue plane. In same way, T_{ru} and T_{rl} are the upper and lower bound respectively for the Red plane. Intersections of these three planes give us ROI.

$\forall \text{ Pixel } (i, j) \in \text{ Test Image},$

Rule 1: $Tsl < \text{Saturation } (i, j) < Tsu,$

$Tsl < Tsu,$

$Tsl, Tsu \in [0, 1]$

Rule 2: $Thl < \text{Hue } (i, j) < Thu$

$Thl < Thu,$

$Thl, Thu \in [0, 1]$

Rule 3: $Trl < \text{Red } (i, j) < Tru,$

$Trl < Tru,$

$Trl, Tru \in [0, 1]$

3.1 Face Tracking Algorithm

Base of the face tracking is derived from our work described in [4]. Frame by frame processing is done on entire video and detected faces are combined to generate a video sequence with notification of facial movement.

Algorithm:

1. Load video Segment Vs
2. For each frame i in Video Segment Vs do
 - If frame (i) satisfy rule 1 to rule 3 than
 - Classify it as a face frame
 - Locate face on frame
 - Else
 - Classify it as a non face frame
 - End
 - Add frame to new video
- End

4 Results and Conclusions

Figure 6 and figure 7 shows the result of face detection algorithm. Faces images used here are of CVL and LFW database. Accuracy of this algorithm is extremely good. In

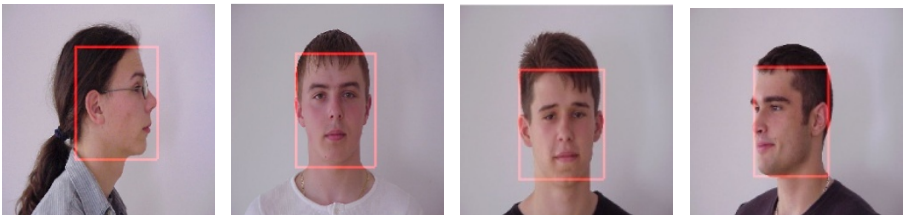


Fig. 6. Results of CVL face database

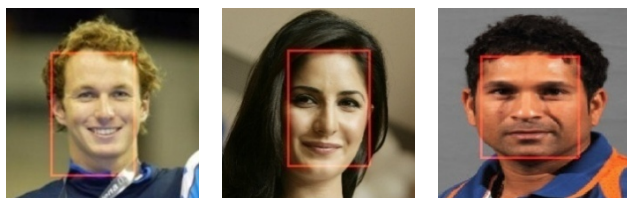


Fig. 7. Results of LFW Database

Table 1. Video Description for Face Tracking

Video Sequence	Video Duration	Total Frames
V1	04.47 Sec.	67
V2	10.80 Sec.	163
V3	07.07 Sec.	106



Fig. 8. Frame Sequences for Video V1



Fig. 9. Frame Sequences for Video V2

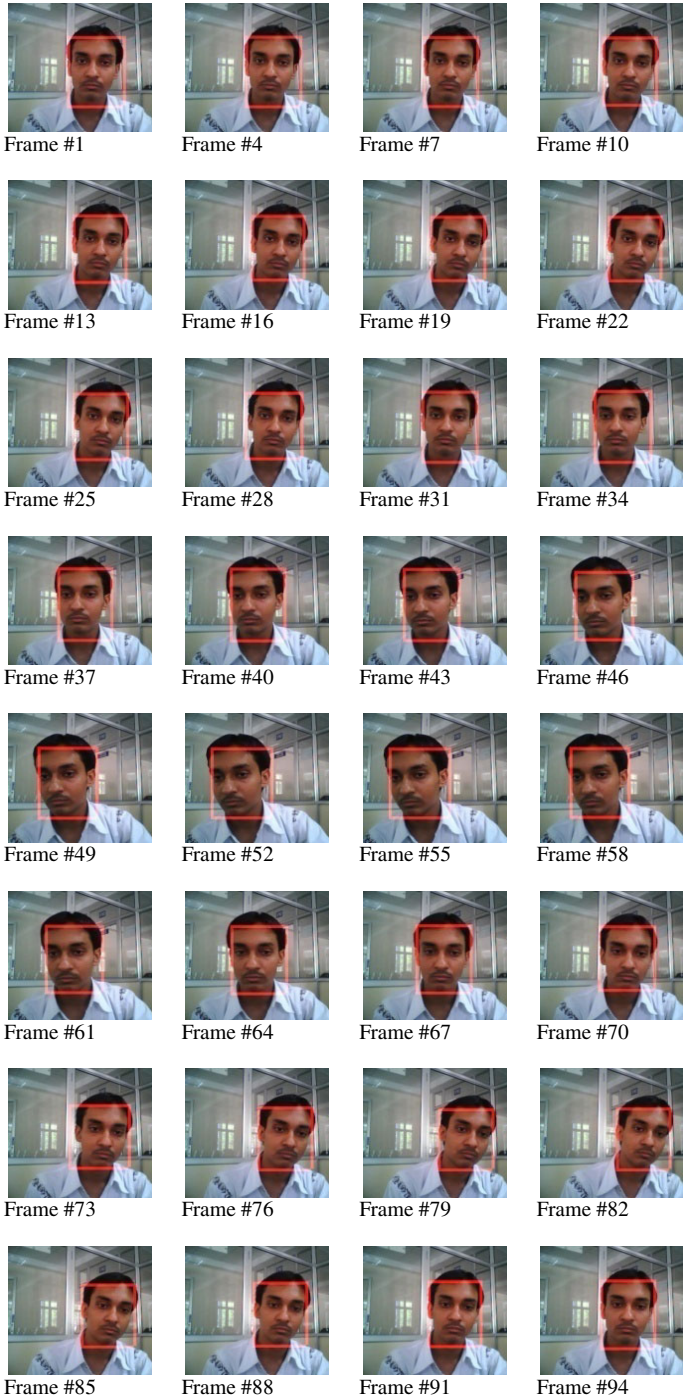


Fig. 10. Resultant Frame Sequences for Video Segment V3

CVL database, faces are of single race and are taken under uniform condition. But in LFW, all images are taken from internet and hence all people are of different region, race and having different face geometry. These images have wide range of intensity, but still this algorithm is able to locate the face region correctly on test image.

Videos used for experiment are recorded on laptop web camera with resolution of 600 X 400 with 15 frames / second. Background was kept complex intentionally to verify the robustness of algorithm. Subject skin colour is also varying heavily. In Video sequence V1, subject skin colour is much yellowish, in video sequence V2, its reddish and in sequence V3 colour of subject face is kind of wheat. Table 1 describes the statistical property of videos used for testing.

Figure 8 and figure 9 show some of the frames of resultant videos. We have generated random number six times for each video from one to number of frames in that video. And frames shown here are selected according to the random number. In short, we have selected arbitrary frames; still all are with very good accuracy. Just for reference, more number of frame sequences is shown in figure 10 at the end of the paper for video sequence V3.

References

- [1] Cooray, S., O'Connor, N.: A hybrid technique for face detection in colour images
- [2] Campadelli, P., Cusmai, F., Lanzarotti, R.: A colour based method for face detection
- [3] Tao, W., Jia-Jun, B., Chun, C.: A colour based face detection system using multiple templates. *Journal of Zhejiang University Science* 4(2), 162–165 (2003)
- [4] Goyani, M., Joshi, B., Shikkenawis, G.: Acceptance / Rejection Rule Based Algorithm for Multiple Face Detection in Colour Images. *International Journal of Engineering Science and Technology* 2(6), 2148–2154 (2010)
- [5] Singh, S.K., Chauhan, D.S., Vatsa, M., Singh, R.: A robust skin colour based face detection algorithm
- [6] Vezhnevets, V., Sazonov, V., Andreeva, A.: A survey on pixel-based skin colour detection techniques
- [7] Deng, X., Chang, C.-H., Brandle, E.: A new method for eye extraction from facial image. In: *IEEE International Workshop on Electronic Design, Test and Applications*, pp. 29–34 (2004)
- [8] Hong, W.-B., Chen, C.-Y., Chang, Y., Fan, C.-H.: Driver fatigue detection based on eye tracking and dynamic template matching. In: *IEEE International Conference on Networking, Sensing & Control*, pp. 7–12 (2003)
- [9] Hsu, R.-L., Abdel-Mottaleb, M., Jain, A.K.: Face detection in colour images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(5), 696–706 (2002)
- [10] Rahman, N.A., Wei, K.C., See, J.: RGB-H-CbCr skin colour model for human face detection.
- [11] Huynh-Thu, Q., Meguro, M., Kaneko, M.: Skin-colour-based image segmentation and its application in face detection
- [12] Fang, J., Qiu, G.: A Colour histogram based approach to human face detection, School of Computer Science, The University of Nottingham
- [13] Kryszczuk, K.M., Drygajło, A.: Colour correction for face detection based on human visual perception metaphor, Signal processing institute, Federal Institute of Technology, Lausanne
- [14] Albiol, A., Torres, L., Delp, E.J.: An unsupervised colour image segmentation algorithm for face detection applications. University of Catalonia and Polytechnic University of Valencia, Spain
- [15] Mohsin, W., Ahmed, N., Mar, C.-T.: Face detection project, Department of Electrical Engineering. Stanford University (2003)

Performance Analysis of Block and Non Block Based Approach of Invisible Image Watermarking Using SVD in DCT Domain

Mahesh Goyani¹ and Gunvantsinh Gohil²

¹ Department of Computer Engineering, GCET, Sardar Patel University,
Anand, Gujarat, India

² Department of Computer Engineering, LDRP, Gujarat University, Gandhinagar, Gujarat, India
mgoyani@gmail.com, gunvantshinh@gmail.com

Abstract. Nowadays image authentication plays a vital role for security. Watermarking techniques facilitate to hide the data in carrier image in such a way so that data would become imperceptible. Watermarking can be performed either in spatial domain or in transfer domain. *Least Significant Bit (LSB)* modification is the simple but non reliable theme, which is performed in spatial domain. In transfer domain, *Singular Value Decomposition (SVD)*, *Discrete Cosine Transformation (DCT)*, *Fast Fourier Transformation (FFT)*, *Discrete Wavelet Transformation (DWT)* and *Fast hadamard Transform (FHT)* etc provide robust result. The goal of the paper is to present how such methods are useful to authenticate images and avoid image tampering. In our work, we have discussed block and non block based watermarking scheme performed in DCT domain using SVD. Several experiments are carried out to check the robustness of the algorithm. We conclude that non block based algorithm is superior in terms of visual quality after embedding and extraction, while its inferior in terms of some common signal processing attacks like rotation, scaling, translation etc.

Keywords: Watermarking, Transform domain, Discrete Cosine Transform, Singular Value Decomposition, Singular Values.

1 Introduction

Watermarking is the process of embedding data into a multimedia element such as image, audio or video [1], [2]. Due to increasing use of internet and communication channels, data security has been a prime concern today. Digital watermarking has received lot of attention in recent time in the area of security. Watermarking techniques are good at providing content authentication and copy right protection for multimedia data. Sensitive data is embedded in to carrier image using various algorithms and at the receiver end data are extracted using appropriate extraction algorithm. Invisibility refers to the degree of distortion introduced by the watermark and its affect on the viewers or listeners [2].

Over the last decade, digital watermarking has become one of the most common ways to deter people copying images without permission. The main goal of watermarking is to hide a message m in some image (*cover image*) data D , to obtain new data D' , practically indistinguishable from D . The data to be embedded in the image is

called *watermark*. A watermark can be classified into two sub-types: visible and invisible. Invisible watermarks operate by embedding data information within the image itself. Invisible watermarks are generally used for copyright protection (robust) and content authentication (fragile).

Robust watermarks are designed to resist attacks to remove or destroy the watermark. Their primary applications are copyright protection. *Fragile watermarks* are designed in such a manner that can be easily destroyed even if the watermarked image is manipulated in the slightest manner.

The general requirement in devising a new watermarking scheme for copyright protection is to achieve a compromise between the invisibility of the hidden watermark, while maintaining a high quality of the digital media, and its robustness against common signal processing operations and specific attacks aimed at removing the watermark or making it undetectable [3]. SVD based watermarking technique in DCT transform domain for colour images are discussed in [4] and [5].

2 Transfer Domain Property

The majority of watermarking techniques can be categorized as algorithms operating either in the spatial domain or in the transform domain [1], [3]. In spatial domain value of image pixels are directly modified and hence it has sometimes adverse effect on colour and brightness of the cover image. Such methods are simple and computationally cheaper but are less robust. Common examples are methods based on the frequency domain, such as Discrete Cosine Transform (DCT) [1], [6], [7], [8], Discrete Fourier Transform (DFT) [9], [10] and Discrete Wavelet Transform (DWT) [11], [12], [13], or based on the Singular Value Decomposition (SVD) [1], [3], [4], [6], [7], [8], [9]. Transform domain are invertible and hence it is easy to modify the transform coefficients and invert the domain. This inverse process spreads the hidden data over entire cover image, so it is bit difficult to extract out it. Transform domain methods allow easier processing of image data, and require less computational power, at the same time providing higher robustness to subsequent manipulation of the marked image. They are generally more robust than those in the spatial domain, since most of the attacks can be characterized and modeled in the transform domain [14].

In all frequency domain watermarking schemes, there is a conflict between robustness and transparency. If the watermark is embedded in perceptually most significant components, the scheme would be robust to attacks but it would be difficult to hide the watermark. On the other hand, if the watermark is embedded in perceptually insignificant components, it would be easier to hide the watermark but the scheme may be less resilient to attacks [1].

2.1 Discrete Cosine Transformation

Transform coding relies on the premise that pixels in an image exhibit a certain level of correlation with their neighboring pixels. Similarly in a video transmission system, adjacent pixels in consecutive frames show very high correlation. Consequently, these correlations can be exploited to predict the value of a pixel from its respective neighbors. A transformation is, therefore, defined to map this spatial (correlated) data into

transformed (uncorrelated) coefficients. Clearly, the transformation should utilize the fact that the information content of an individual pixel is relatively small i.e., to a large extent visual contribution of a pixel can be predicted using its neighbors [15].

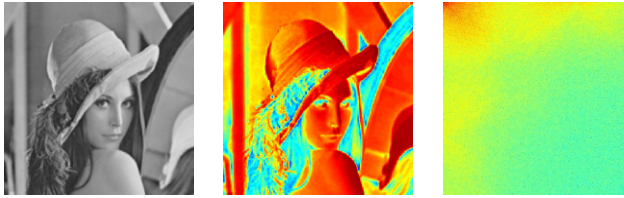


Fig. 1. (a). Test image 'Lena' (b). Energy map of Test image (c). Energy map of DCT transformed Test image.

Fig. 1 (a) shows the test image Lena. jpg. Energy map of the test image is shown in fig. 1(b). It is clear from the figure that energy of the signal is distributed over entire image. DCT transferred test image is shown in fig. 1(c), where most of the energy of the signal is concentrated in initial parameters. Reading these parameters in zigzag order gives the most significant to least significant coefficients of DCT domain.

2.2 Singular Value Decomposition

The SVD for square matrices was discovered independently by Beltrami in 1873 and Jordan in 1874, and extended to rectangular matrices by Eckart and Young in the 1930s. It was not used as a computational tool until the 1960s because of the need for sophisticated numerical techniques. In later years, Gene Golub demonstrated its usefulness and feasibility as a tool in a variety of applications. SVD is one of the most useful tools of linear algebra with several applications in image compression, watermarking and other signal processing fields [2]. The SVD is a transform suitable for image compression because it provides optimal energy compaction for any given image [16]. But its application is very limited due to the computational complexity associated with the computation of eigenvalues and eigenvectors [17]. A good representation of the image can be achieved by taking only a few largest eigenvalues and corresponding eigenvectors [1], [16], [17].

SVD based hybrid techniques works in transform domain which enhances the robustness and resist to different kind of attacks. Embedding in low frequencies increases the robustness with respect to attacks that have low pass characteristics like filtering, lossy compression, and geometric distortions while making the scheme more sensitive to modifications of the image histogram, such as contrast/brightness adjustment, gamma correction, and histogram equalization. Watermarks embedded in middle and high frequencies are typically less robust to low-pass filtering, lossy compression and small geometric deformations of the image but are highly robust with respect to noise adding, and nonlinear deformations of the gray scale.

Advantages and disadvantages of low and middle-to-high frequency watermarks are complementary [2]. Considering these property visual watermarks are embedded in one image in four sub bands. In recent years, SVD has gained lot of attraction because of its attractive features like simplicity in implementation and mathematical proof. SVD is well known technique for factorizing a rectangular matrix. This decomposition of matrix is very much useful in signal processing like image coding,

noise reduction, watermarking etc. SVD transformation was used to measure image quality under different types of distortion. In SVD, original matrix of size $m \times n$ is decomposed in three matrices $[U \Sigma V^T]$ of size $m \times m$, $m \times n$ and $n \times n$ respectively. If the source matrix is square matrix of size n than U , Σ and V are of same size that is $n \times n$. Image is an array of non negative scalar value and hence the decomposition of image array A would be,

$$SVD(A_{m \times n}) = [U_{m \times m} \Sigma_{m \times n} V_{n \times n}] \tag{1}$$

Where, U and V are orthogonal matrices. So,

$$U \cdot U^T = I_{\text{identity}(m \times m)}$$

$$V \cdot V^T = I_{\text{identity}(n \times n)}$$

U and V^T , represents eigen vector of A , and a diagonal matrix Σ represents eigen values of A . Σ is matrix with all non diagonal values zero.

$$A = U \Sigma V^T \tag{2}$$

$$\Sigma = \text{diag}(\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_r, 0, 0, 0, \dots)$$

$$\Sigma = \text{diag}(\sigma_1^2, \sigma_2^2, \sigma_3^2, \dots, \sigma_r^2, 0, 0, 0, \dots)$$

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_r > \lambda_{r+1} = \dots = \lambda_n = 0$$

Where, r is the rank of matrix A . Columns $v(n)$ of V can be calculated as,

$$(A' - \lambda(n)I)v(n) = 0, \quad n = 1, 2, \dots, r \tag{3}$$

Where, $A' = A^T A$
Columns of U are,

$$u(n) = \frac{1}{\sqrt{\lambda(n)}} Av(n), \quad n = 1, 2, \dots, r \tag{4}$$

$A = 1 \dots \min(m, n)$ are the singular values, sorted in decreasing order in diagonal of S . This decomposition is known as the Singular Value Decomposition (SVD) of A , and can be written as,

$$A = \sigma_1 U_1 V_1 + \sigma_2 U_2 V_2 + \dots + \sigma_r U_r V_r \tag{5}$$

To calculate the SVD, we need to calculate eigenvalues eigenvectors of $A \cdot A^T$ and $A^T \cdot A$. Eigenvectors of $A \cdot A^T$ forms the columns of U , while Eigenvectors of $A^T \cdot A$ forms the columns of V . And the square roots of eigenvalues of $A^T \cdot A$ or $A \cdot A^T$. The columns of U are called the left singular vectors of A , and the columns of V are called the right singular vectors of A .

It is important to note that each singular value specifies the luminance of an image layer while the corresponding pair of singular vectors specifies the geometry of the image [3]. Very interesting property of SVD is that magnitude of SVs directly depends on the image luminance. Increasing the magnitude of singular values increase the luminance of image while lowering the magnitude of singular values decreases the image luminance. Intrinsic geometric property of image depends on U and V , which are responsible for horizontal and vertical details respectively [3]. SVD is the invariance of SVs to common image processing operations (except for noise addition) and geometric

transforms, like rotation, translation, scaling etc. [10]. SVD packs the maximum signal energy into as few coefficients as possible. SVs of image have very good property of stability, means if small jitter is added to image, SVs do not very sharply.

There are many SVD based algorithms has been developed so far for the image watermarking like,

- Modify singular values of host image [19].
- Modify singular vectors of host image.
- Modify singular vectors and values of host image [19], [20].
- Combine above features with other transform domain (DFT, DCT, FFT etc.)

3 Proposed Algorithm

SVD based algorithm can be classified in two categories. Pure SVD based algorithms and SVD and transform domain based algorithm [1]. DCT, DWT, DFT, FHT, Haar transformation etc. transform domain can be combined with SVD based algorithm. In this paper we presented comparison between block-based and non block based watermarking scheme, which uses the Singular Value Decomposition transform in DCT domain. The latter tends to spread the watermark all over the image, whereas the former only affects local regions of the image [3].

3.1 Block Based Approach

The proposed scheme works by initially splitting the original image into non-overlapping blocks, applying the SVD transform to each of them and subsequently embedding a watermark into the singular values. Each watermark value is embedded by modifying a set of singular values. In [3], author has discussed the same approach, but they have modified Singular Vectors. Before embedding, the watermark image is reduced to half of its actual size to fit in each block of the cover image. In our experiment, we have considered size of cover image 256 X 256 and size of watermark 128 X 128 pixels. Fig. 2 and fig. 3 shows the flow of watermark embedding and extraction procedure.

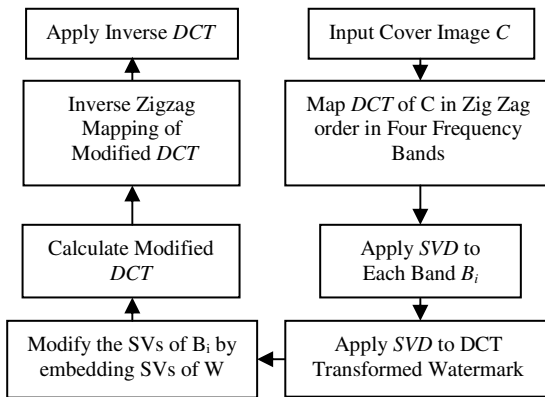


Fig. 2. Watermark embedding process

According to the algorithm the cover image is divided into four blocks and the watermark image is embedded in each block. In the extraction phase, all four watermark images from each block are extracted. Quality of the extracted watermark is evaluated visually and using the Pearson correlation coefficient and many other similarity measurement factors between the original watermark image and extracted watermark image from each block. In this approach watermark image is hidden in four quadrants and hence extraction process returns four watermarks.

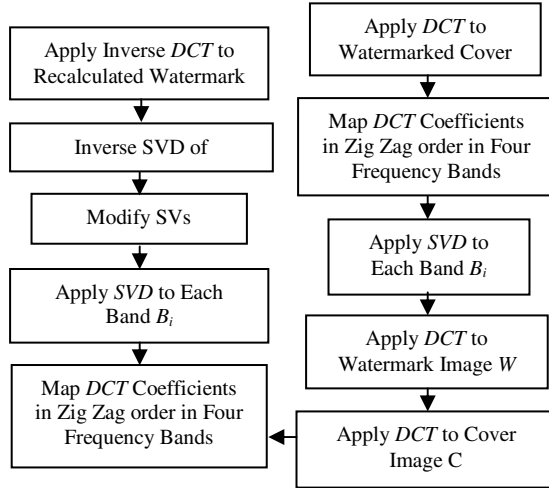


Fig. 3. Watermark extraction process

3.2 Non Block Based Approach

Many of the earlier algorithms, based on SVD, used to embed the watermark signal directly into the SVD domain. In non block based algorithm, cover image and watermark image are of same size. We have considered dimension of 256 X 256. Algorithmic steps are same as previous case, only difference is that we are not dividing the cover image in sub blocks. In block based approach, watermark is embedded four times in cover image, while in this approach whole watermark is embedded once over entire cover image.

4 Results and Conclusions

This watermarking scheme is RST invariant due to its robustness against rotation, translation and scaling attacks. Moreover it is resilient to many other attacks including Gaussian Blur, Gaussian noise, JPEG compression, histogram equalization etc. The cover image used in this experiment is ‘Lena’ of size 256 × 256 and the watermark image is ‘Cameraman’ of the size 128 X 128.



Fig. 4. Cover image Lena and extracted results of all four quadrants

Most significant DCT coefficients are in quadrant one and have less significance in remaining quadrants in decreasing order. Extracted watermark results are shown in figure 4. Pearson coefficient is good measurement of similarity in image. In Table 1, results are discussed. Resemblance between original watermark and extracted one is 0.99977, shows high degree of retrieval quality of algorithm. In the same sense, resemblance between original cover image and watermarked image is also very high, near to 1 (0.99932 exactly).

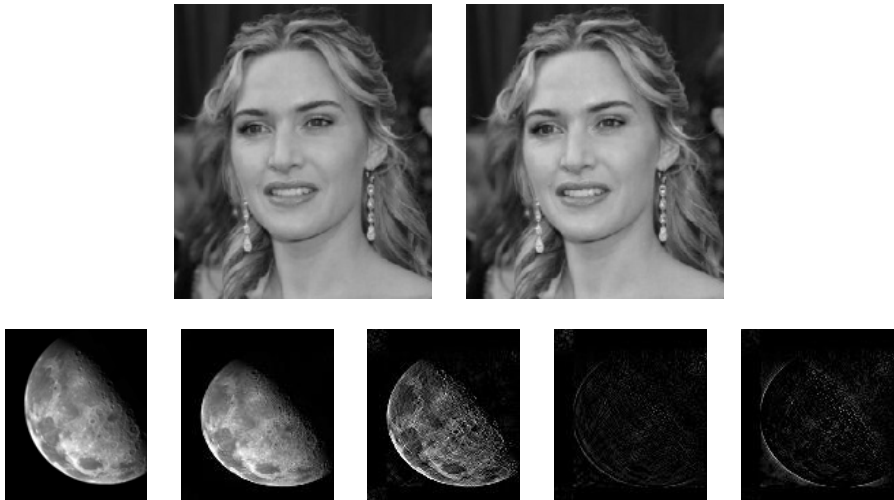


Fig. 5. (a). Top Row from left: Cover image and watermarked image (b). Bottom row from left: Original watermark and extracted watermarks from four quadrants.

In figure 5, we can see that luminance of original cover image has been changed slightly. But it still preserves good visual property, so that no one can think about tempering of cover image. Second row of figure 5 shows original watermark and extracted watermarks of all four quadrants.

Table 1. Performance Matrix (Block based approach)

Image	Cover Image	Watermark	Cover Image	Watermark
Parameter	Lena	Cameraman	Cat	Moon
Pearson Coefficient	0.99932	0.99977	0.99971	0.99983
MSE	101.797	1.825	34.084	1.381
PSNR	64.595	104.811	75.537	107.598

Statistics shows that non block based watermarking approach gives better results compared to block based approach. For non block based algorithm, size of cover image and watermark are same i.e. 256 X 256. Performance matrix in table 3 and table 4 clearly shows that non block based approach preserves high visual quality compared to block based approach. Pearson coefficient for block and non block based approach is 0.99932, 0.99977 and 0.99999 and 1.0 for cover and watermark image respectively. Fig. 6 and fig. 7 describes the results for non block based approach.



Fig. 6. (a). Cover image 'lena.jpg'. (b). Watermarked image. (c). Watermark 'Cameraman.tif'. (d). Extracted watermark.



Fig. 7. (a). Cover image 'cat.jpg'. (b). Watermarked image. (c). Watermark 'Moon.tif'. (d). Extracted watermark.

Table 2. Performance Matrix (Non Block based approach)

Image	Cover Image	Watermark	Cover Image	Watermark
Parameter	Lena	Cameraman	Cat	Moon
Pearson Coefficient	0.99999	1.00000	0.99999	1.00000
MSE	0.0178	0.0000	0.006	0.0000
PSNR	151.107	337.891	162.025	627.675

Some other error measurement properties are also exploited to verify the quality of algorithm. Lower the Mean Square Error (MSE), higher the originality in signal. And higher the Peak Signal to Noise Ratio (PSNR), higher the originality in signal. Results it self speaks that MSE is almost zero for non block based approach.

References

- [1] Rfizul Haque, S.M.: Singular Value Decomposition and Discrete Cosine Transformation Based Image Watermarking. Master Thesis, Dept. of Interaction and System Design, Blekinge Institute of Technology, Sweden (January 2008)
- [2] Sverdllov, A., Dexter, S., Eskicioglu, A.M.: Secure DCT-SVD Domain Image Watermarking: Embedding Data in All Frequencies
- [3] Basso, A., Bergadano, F., Cavagnino, D., Pomponiu, V., Vernone, A.: A Novel Block-based Watermarking Scheme Using the SVD Transformation
- [4] Wu, H.-C., Yeh, C.-P., Tsai, C.-S.: A Semi-fragile Watermarking Scheme Based on SVD and VQ Techniques. In: Gavrilova, M.L., Gervasi, O., Kumar, V., Tan, C.J.K., Taniar, D., Laganá, A., Mun, Y., Choo, H. (eds.) ICCSA 2006. LNCS, vol. 3982, pp. 406–415. Springer, Heidelberg (2006)
- [5] Yavuz, E., Telatar, Z.: SVD Adapted DCT Domain DC Sub-band Image Watermarking Against Watermark Ambiguity. In: Günsel, B., Jain, A.K., Tekalp, A.M., Sankur, B. (eds.) MRCS 2006. LNCS, vol. 4105, pp. 66–73. Springer, Heidelberg (2006)
- [6] Cox, I., Kilian, J., Leighton, F.T., Shamoon, T.: Secure spread spectrum watermarking for multimedia. In: IEEE Transaction on Image Processing, Piscataway, New Jersey, USA, pp. 1673–1687 (December 1997)
- [7] Barni, M., Bartolini, F., Piva, A.: A DCT domain system for robust image watermarking. IEEE Transactions on Signal Processing 66, 357–372 (1998)
- [8] Chu, W.C.: DCT based image watermarking using sub Sampling. IEEE Trans Multimedia 5, 34–38 (2003)
- [9] Ruanaidh, J.J.K., Dowling, W.J., Boland, F.M.: Phase watermarking of digital images. In: Proceedings of the 1996 International Conference on Image Processing, Lausanne, Switzerland, vol. 3, pp. 239–242 (September 1996)
- [10] Peiningand, T., Eskicioglu Ahmet, M.: An Adaptive Method for Image Recovery in the DFT Domain. Journal of Multimedia 1(6) (September 2006)
- [11] Dugad, R., Ratakonda, K., Ahuja, N.: A new wavelet-based scheme for watermarking images. In: International Conference on Image Processing Proceedings (ICIP 1998), Chicago, USA, October 4-7, vol. 2, pp. 419–423 (1998)
- [12] Barni, M., Bartolini, F., Piva, A.: Improved wavelet based watermarking through pixel-wise masking. IEEE Trans Image Processing 10, 783–791 (2001)
- [13] Wang, Y., Doherty, J.F., Van Dyck, R.E.: A wavelet based watermarking algorithm for ownership verification of digital images. IEEE Transactions on Image Processing 11(2), 77–88 (2002)
- [14] Podilchuk, C., Zeng, W.: Image-adaptive watermarking using visual models. IEEE J. Sel. Are. Comm. 16, 525–539 (1998)
- [15] Khayam, S.A.: The Discrete Cosine Transform – Theory and Application, Department of Electrical and Computer engineering, Michigan State university (2003)
- [16] Ochoa, H., Rao, K.R.: A Hybrid DWT-SVD Image-Coding System (HDWTSVD) for Color Images

- [17] Jain, A.K.: Fundamentals of Digital Image Processing. Prentice-Hall, Englewood Cliffs (1989)
- [18] Calagna, M., Guo, H., Mancini, L.V., Jajodia, S.: Robust Watermarking System based on SVD Compression. In: Proceedings of the 2006 ACM Symposium on Applied Computing, Dijon, France, April 23 -27, pp. 1341–1347 (2006)
- [19] Chang, C.-C., Tsai, P., Lin, C.-C.: SVD-based Digital Image Watermarking scheme. Pattern Recognition Letters 26, 1577–1586 (2005)
- [20] Mohan, B.C., Kumar, S.: A Robust Digital Image Watermarking Scheme using Singular Value Decomposition. J. Multimed. 3, 7–15 (2008)

A Novel Topographic Feature Extraction Method for Indian Character Images

Soumen Bag and Gaurav Harit

Indian Institute of Technology Kharagpur, Kharagpur-721302, India
{soumen,gharit}@cse.iitkgp.ernet.in

Abstract. In this paper, we present novel features based on the topography of a character as visible from different viewing directions on a 2D plane. By topography of a character we mean the structural features of the strokes and their spatial relations. In this work we develop topographic features of strokes visible with respect to views from different directions (e.g. North, South, East, and West). We consider three types of topographic features: closed region, convexity of strokes, and straight line strokes. We have tested the proposed method on printed and handwritten Bengali and Hindi isolated character images. Initial results demonstrate the efficacy of our approach.

Keywords: Convexity, Indian script, OCR, Thinning, Topographic feature.

1 Introduction

Feature selection and extraction has wide range of application for pattern recognition. It plays an important role in different classification based problems such as face recognition, signature verification, optical character recognition (OCR) etc.. From past several decades, many feature selection and extraction methods are reported [19] for Indian character recognition. The accuracy rate of OCR depends on feature sets and classifiers [16]. Next we begin a brief description about few important feature sets used in optical character recognition for Bengali and Hindi documents.

Chaudhuri and Pal [5] proposed the first complete printed Bengali OCR in 1998. In this method, nine different strokes are used as primary feature set for recognizing basic characters and template-based features are used for recognizing compound characters. To recognize handwritten basic and compound characters, Das *et al.* [6] used shadow, longest run, and quad-tree based features. Dutta and Chaudhuri [7] proposed topological features, such as junction points, hole, stroke segments, curvature maxima, curvature minima, and inflexion points of character images for performing printed and handwritten Bengali alpha-numeric character recognition. To detect convexity of Bengali numerals, Pal and Chaudhuri [13] used water-flow model. They also used topological and statistical features for preparing feature set. Bhowmick *et al.* [4] proposed a stroke-based feature set for recognizing Bengali

handwritten character images. In this method, ten stroke based features which indicate the shape, size and position information of a digital curve with respect to the character image, are extracted from character images to form the feature vector. Majumdar [11] have introduced a new feature extraction method based on the curvelet transform of morphologically altered versions of an original character image. Table 1 gives a summary of different feature sets used in Bengali OCR systems.

Table 1. Different feature sets used in Bengali OCR systems

Method	Feature set
Chaudhuri and Pal [5]	Structural and template features
Das <i>et al.</i> [6]	Shadow, longest run and quad-tree based features
Dutta and Chaudhuri [7]	Structural and topological features
Pal and Chaudhuri [16]	Watershed, topological, and statistical features
Bhowmick <i>et al.</i> [4]	Stroke based feature
Majumdar [11]	Curvelet coefficient features

The first complete OCR on Devanagari was introduced by Pal and Chaudhuri [12]. In this method, they have used structural and template features for recognizing basic, modified, and compound characters. To recognize real-life printed documents of varying size and font, Bansal and Sinha [3] proposed statistical features. Later Pal *et al.* [15] used the same gradient features for recognizing handwritten Devanagari characters. Bajaj *et al.* [2] used density, moment of curve and descriptive component for recognizing Devanagari handwritten numerals. Sethi and Chatterjee [17] proposed a set of primitives, such as, global and local horizontal and vertical line segments, right and left slant, and loop for recognizing handwritten Devanagari characters. Sharma *et al.* [18] used directional chain code information of the contour points of the characters for recognizing handwritten Devanagari characters. Table 2 gives a summary of different feature sets used in Devanagari OCR systems.

Table 2. Different feature sets used in Devanagari OCR systems

Method	Feature set
Pal and Chaudhuri [12]	Structural and template features
Bansal and Sinha [3]	Statistical features
Bajaj <i>et al.</i> [2]	Density, moment of curve and descriptive component features
Sethi and Chatterjee [17]	Line segments, slant, and loop
Pal <i>et al.</i> [15]	Gradient features
Sharma [18]	Directional chain code information of contour points

All of the above mentioned methods do not consider shape variation for extracting features. But in Indian languages, a large number of similar shape type characters (basic and conjunct) are present. From that point of view, we have proposed novel features based on the topography of a character to improve the performance of existing OCR in Indian script, mainly for Bengali and Hindi documents. The major features of the proposed method are listed as follows:

1. The main challenge to design an OCR for Indian script is to handle large scale shape variation among different characters. Strokes in characters can be decomposed into segments which are straight lines, convexities or closed boundaries (hole). In our work we consider the topography of character strokes from 4 viewing directions. In addition to the different convex shapes formed by the character strokes, we also note the presence of closed region boundaries.
2. The extracted features are represented by a shape-based graph where each node contains the topographic feature, and they all are placed with respect to their centroids and relative positions in the original character image.
3. This approach is applicable for printed as well as handwritten text documents of different languages.

This paper is organized as follows. Section 2 describes the proposed topographic features extraction method. Section 3 contains the experimental results. This paper concludes with some remarks on the proposed method and future work in Section 4.

2 Proposed Topographic Feature Extraction Method

In this section we describe our proposed feature extraction method based on measurement of convexity of character strokes from different directions.

2.1 Preprocessing

Given a scanned document page we binarize it using the Otsu's algorithm [8]. Currently we are working with documents with all text content. For Bengali or Hindi documents the entire word gets identified as a single connected component because of the *mātrā/shiro-rekhā* (head line) which connects the individual characters. For this case we separate out the individual *aksharā* within a word by using the character segmentation methods reported in [14] (for Bengali) and [10] (for Hindi).

2.2 Thinning to Get Skeletonized Image

Before extracting the features, character images are converted to single pixel thick images. But to retain the proper shape of thinned character images is a big challenge. Lot of works have been done on thinning. But most of the works are reported for English, Chinese, and Arabic languages [9]. From this point of view, Bag and Harit have proposed an improved medial-axis based thinning strategy for Indian character images. Details of the methodology is reported in [1]. Few results are shown in Fig. 1.

2.3 Topographic Features Extraction

Topographic features are classified into three categories: closed region, convexity of strokes, and straight line strokes. The convexity of curve is detected from different directions. Here we consider four directions (North, South, East, and West) for convexity measurement.

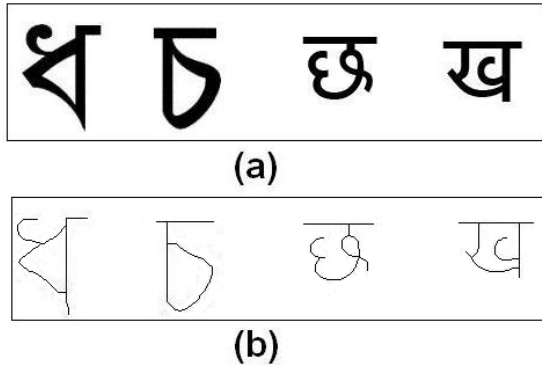


Fig. 1. Skeletonization results: (a) Input image; (b) Skeleton image

Definition of convexity: The convexity of a shape is detected by checking its convergence towards a single point or a cluster of points, connected by 4-connectivity. For any convexity shape, if we move down word then we shall reach to a single point or a flat region which is a set of 4-connected pixels. For detecting the stroke convexity, we have used this concept. Fig. 2 shows the different convexities of character images detected from different view directions. In the figure, different colors are used to mark convexities detected from different view directions (Red, Blue, Green, and Magenta for North, South, East, and West respectively).

The following steps discuss the methodology to detect convexity from North direction.

1. Prepare a database containing different convex shape. Each convex shape is defined according to their structural property. Fig. 3 shows different shapes stored in the database.

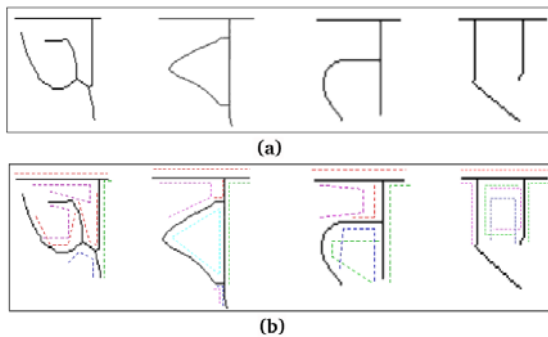


Fig. 2. (a) Skeleton image; (b) Different convex shapes marked by different colors (Red, Blue, Green and Magenta for North, South, East, and West respectively)

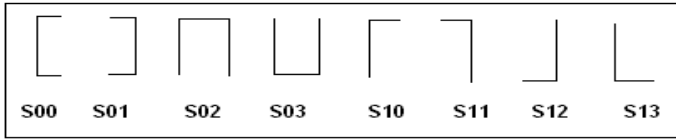


Fig. 3. Different convex shapes

2. Detect horizontal-start (H_s), horizontal-end (H_e), vertical-start (V_s), and vertical-end (V_e) points of the thinned image. Here $[H_s, H_e]$ and $[V_s, V_e]$ are the horizontal and vertical limits of the input image respectively (see Fig. 4).
3. Given a thinned binary image, it is scanned from top to bottom and left to right, and transition from white (background) to black (foreground) are detected. The whole scanning is done from H_s to H_e and V_s to V_e in horizontal and vertical directions respectively.
4. Suppose, a single scan from left to right generates a sequence of pixels $\langle x_1, x_2, x_3, \dots, x_i, x_{i+1}, \dots, x_n \rangle$ where each pixel is a cut point between horizontal scan line and character image. Now, if the two consecutive pixels (x_i & x_{i+1}) are not 4-connected neighbors, then put them in an array of points $P[1 : N, 1 : 2]$ where each cell $P[i]$ contains one pixel with its x, y coordinates in $P[i][1]$ and $P[i][2]$ respectively.
5. Continue the above steps for all remaining horizontal scan until we get a single point or a set of horizontally 4-connected neighbor pixels with value $\geq \zeta$ (set to 5).
6. The detected set of pixels are matched with the defined convex shapes stored in the database to get a specific convex shape.
7. The above steps are repeated for detecting the convexity from remaining three directions, i.e. South, East, and West. Only difference is that, for South direction, the scan is done from bottom to top and left to right, and for East/West directions, scan is done from top to bottom and right to left (for East)/left to right (for West) directions. Finally, a set of different convex shapes are generated to prepare topographic feature set.

The above steps are used for detecting convexity of character strokes from different directions. Now for detecting closed region, we use the concept of connected

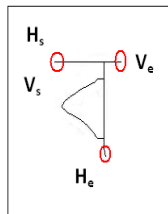


Fig. 4. Horizontal and vertical limits of an image

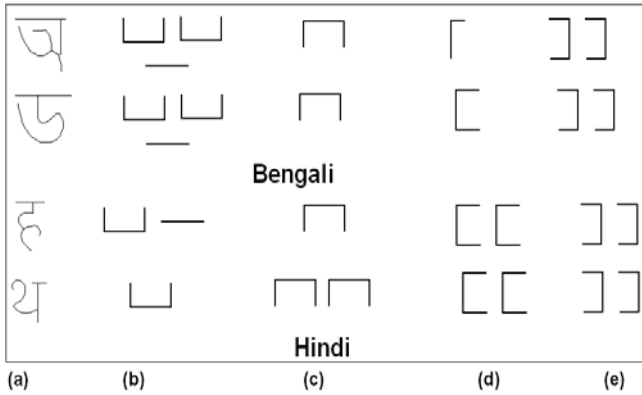


Fig. 5. Topographic features of thinned character images: (a) Skeleton image; (b)-(e) Features from North, South, East, and West direction

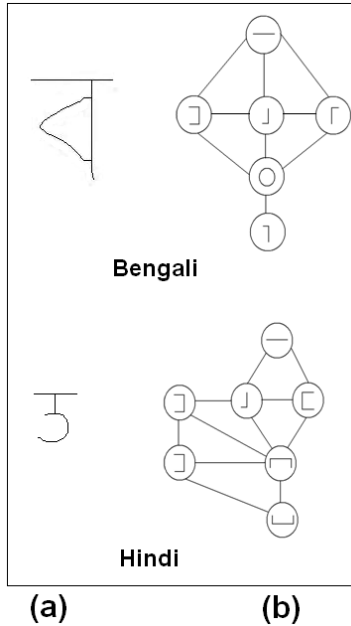


Fig. 6. (a) Skeleton image; (b) Shape-based graph

component analysis. If the pixels of the stroke segment are connected (i.e. each pixel has two 8-connected neighbors), then the stroke is identified as **closed region**.

If the number of horizontally 8-connected neighbor pixels is $\geq \Omega$ (set to 20), then we can say that **straight line** is present. For Bengali and Hindi language, most of the characters have headline. By observing the structural shape of these two type of characters, we can say that the straight line will detect only from

North direction. For this reason, we do not use straight detection method for other three directions.

Fig. 5 shows the different topographic features extracted from Bengali and Hindi thinned character images.

2.4 Graphical Representation of Topographic Features

After detecting the convexity from four directions, we design a shape-based graph to represent the feature set of a particular character. The steps are given below.

1. Suppose, a thinned character image has k number of topographic components $\langle T_1, T_2, \dots, T_k \rangle$.
2. For each component, calculate the centroid (X_i, Y_i) using equation 1.

$$X_i = \frac{\sum_{j=1}^{N_i} p_{ij}}{N_i} ; \quad Y_i = \frac{\sum_{j=1}^{N_i} p'_{ij}}{N_i} \tag{1}$$

where N_i is the total number of black pixels of the i^{th} topographic component, p_{ij} and p'_{ij} are the x-coordinate and y-coordinate of the i^{th} pixel of the j^{th} topographic component respectively.

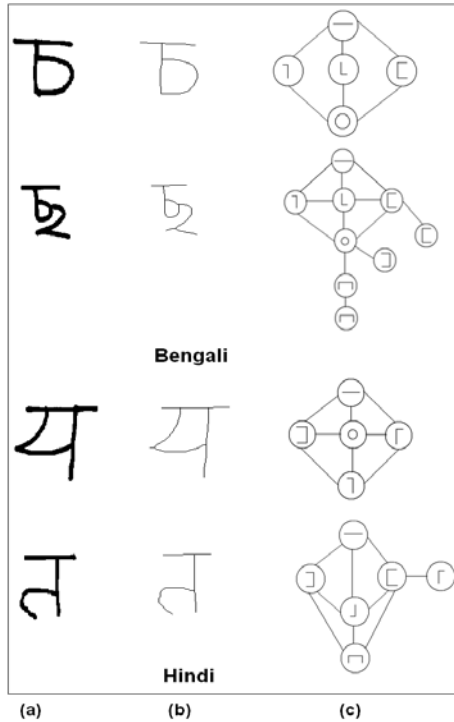


Fig. 7. Topographic feature extraction of handwritten character images: (a) Input image; (b) Skeleton image; (b) Shape-based graph

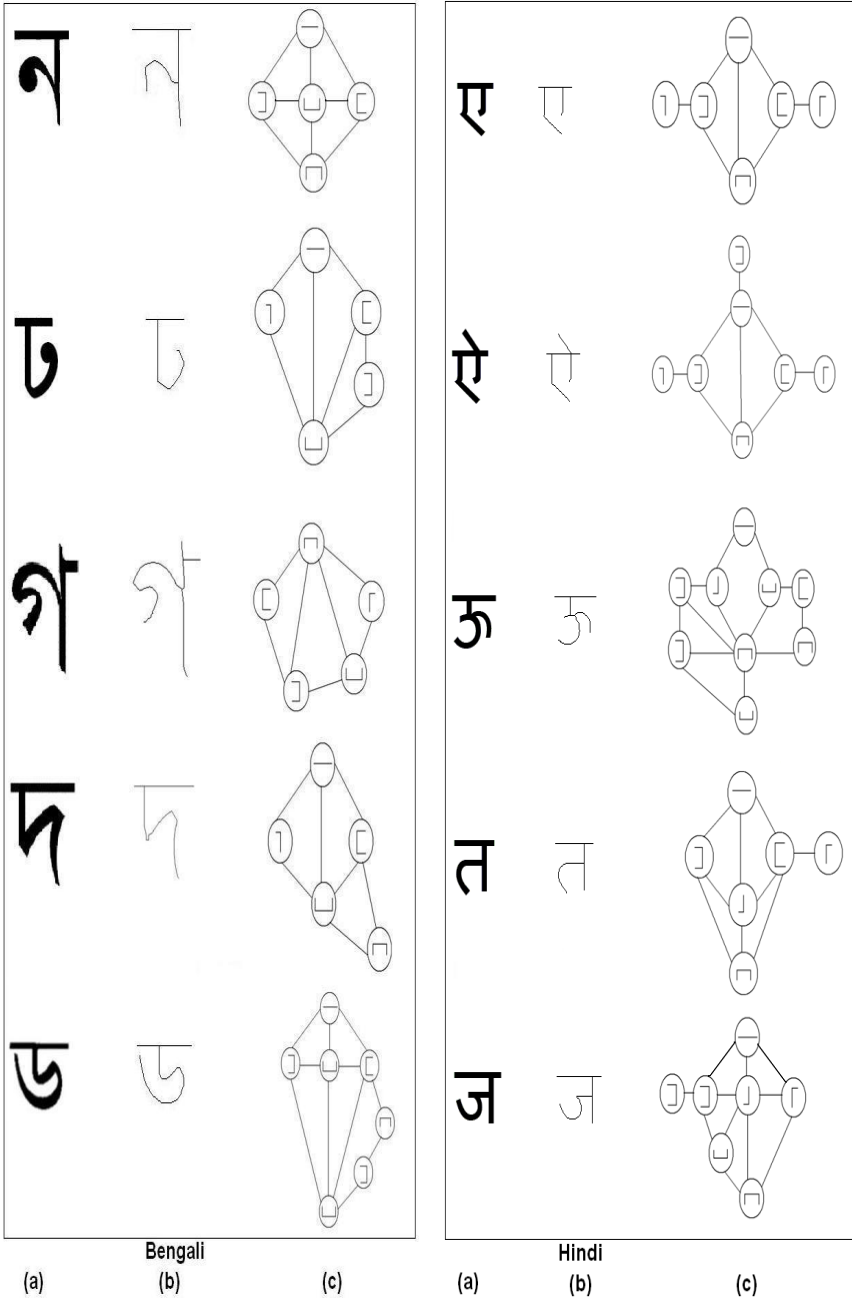


Fig. 8. Topographic features of printed character images: (a) Input image; (b) Skeleton image; (c) Shape-based graph

- Design an undirected graph $G = (V, E)$, where V is the set of vertices containing different topographic features and E is the set of edges. All vertices are placed based on the coordinates of their centroid. Now add edges among these vertices with respect to their relative positions in the thinned character image.

Fig. 6 shows the shape-based graph of thinned Bengali and Hindi character images. This graphical model gives a clear pictorial difference among different feature set of different character images.

3 Experimental Results

We collected characters from several heterogeneous printed and handwritten documents of Bengali and Hindi. The number of characters in the testing set is 1250 (600 for Bengali and 650 for Hindi). All the characters are collected in a systematic manner from printed and handwritten pages scanned on a HP scanjet 5590 scanner at 300 dpi. At first, images were thinned by Bag and Harit thinning method [1]. This thinning method has the capability to preserve the shape of character images at junction points and end points. Then we tested our proposed feature extraction method on these thinned images to get topographic features. Fig. 8 shows the shape-graph representation of topographic features of printed Bengali and Hindi character image. The algorithm is implemented in C++ programming language using OpenCV 2.0 on Unix/Linux platform.

We applied this proposed method on handwritten Bengali and Hindi character images. Fig. 7 shows few results. It is observed that the performance is satisfactory for handwritten characters also.

4 Conclusion

In this paper, we have proposed a novel topographic feature extraction method for Indian OCR systems. This feature set captures close region, convexity of stroke from different direction, and flat region of thinned character images. The proposed method is tested on printed and handwritten Bengali and Hindi documents and we have obtained promising results. The proposed feature set helps to discriminate two similar type characters properly. In future, we shall extend our work to extract topographic features for other popular Indian languages and make it as a script independent feature set for designing multi-lingual OCR in Indian scripts.

References

- Bag, S., Harit, G.: A medial axis based thinning strategy for character images. In: Proceedings of the 2nd National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics, Jaipur, India, pp. 67–72 (2010)
- Bajaj, R., Dey, L., Chaudhury, S.: Devnagari numeral recognition by combining decision of multiple connectionist classifiers. *Sadhana* 27, 59–72 (2002)

3. Bansal, V., Sinha, R.M.K.: Integrating knowledge sources in Devanagari text recognition system. *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans* 30(4), 500–505 (2000)
4. Bhowmick, T.K., Bhattacharya, U., Parui, S.K.: Recognition of Bangla handwritten characters using an MLP classifier based on stroke features. In: *Proceedings of the 11th International Conference on Neural Information Processing*, Kolkata, India, pp. 814–819 (2004)
5. Chaudhuri, B.B., Pal, U.: A complete printed Bangla OCR system. *Pattern Recognition* 31(5), 531–549 (1998)
6. Das, N., Das, B., Sarkar, R., Basu, S., Kundu, M., Nasipuri, M.: Handwritten Bangla basic and compound character recognition using MLP and SVM classifier. *Journal of Computing* 2(2), 109–115 (2010)
7. Dutta, A., Chaudhury, S.: Bengali alpha-numeric character recognition using curvature features. *Pattern Recognition* 26(12), 1757–1770 (1993)
8. Gonzalez, R.C., Woods, R.E.: *Digital Image Processing*. Prentice Hall, USA (2008)
9. Lam, L., Lee, S.W., Suen, C.Y.: Thinning methodologies—A comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14(9), 869–885 (1992)
10. Ma, H., Doermann, D.: Adaptive Hindi OCR using generalized hausdorff image comparison. *ACM Transactions on Asian Language Information Processing* 2, 193–218 (2003)
11. Majumdar, A.: Bangla basic character recognition using digital curvelet transform. *Journal of Pattern Recognition Research* 2(1), 17–26 (2007)
12. Pal, U., Chaudhuri, B.B.: Printed Devnagari script OCR system. *Vivek* 10, 12–24 (1997)
13. Pal, U., Chaudhuri, B.B.: Automatic recognition of unconstrained off-line Bangla handwritten numerals. In: *Proceedings of the 3rd International Conference on Advances in Multimodal Interfaces*, Beijing, China, pp. 371–378 (2000)
14. Pal, U., Datta, S.: Segmentation of Bangla unconstrained handwritten text. In: *Proceedings of the 7th International Conference on document Analysis and Recognition*, Edinburgh, Scotland, pp. 1128–1132 (2003)
15. Pal, U., Sharma, N., Wakabayashi, T., Kimura, F.: Offline handwritten character recognition of Devnagari script. In: *Proceedings of the 9th International Conference on Document Analysis and Recognition*, Curitiba, Brazil, pp. 496–500 (2007)
16. Pal, U., Wakabayashi, T., Kimura, F.: Comparative study of Devnagari handwritten character recognition using different feature and classifiers. In: *Proceedings of the 10th International Conference on Document Analysis and Recognition*, Barcelona, Spain, pp. 1111–1115 (2009)
17. Sethi, K., Chatterjee, B.: Machine recognition of constrained hand-printed Devnagari. *Pattern Recognition* 9, 69–77 (1977)
18. Sharma, N., Pal, U., Kimura, F., Pal, S.: Recognition of offline handwritten Devnagari characters using quadratic classifier. In: *Proceedings of the 5th Indian Conference on Computer Vision, Graphics and Image Processing*, Madurai, India, pp. 805–816 (2006)
19. Trier, O.D., Jain, A.K., Taxt, T.: Feature extraction methods for character recognition—A survey. *Pattern Recognition* 29(4), 641–662 (1996)

Comprehensive Framework to Human Recognition Using Palmprint and Speech Signal

Maresh P.K.^{1,*} and M.N. ShanmukhaSwamy²

¹ Research Scholar, Department of Electronics and Communication, J.S.S. Research Foundation, Mysore University, Mysore
mahesh24pk@gmail.com

² Professor, Department of Electronics and Communication, J.S.S. Research Foundation, Mysore University, Mysore
mnsjce@gmail.com

Abstract. This paper presents fusion of two biometric traits, i.e., palmprint and speech signal, at matching score level architecture uses weighted sum of score technique. The features are extracted from the pre-processed palm image and pre-processed speech signal. The features of a query image and speech signal are compared with those of a database images and speech signal to obtain matching scores. The individual scores generated after matching are passed to the fusion module. This module consists of three major steps i.e., normalization, generation of similarity score and fusion of weighted scores. The final score is then used to declare the person as genuine or an impostor. The system is tested on database collected by the authors for 120 subjects and gives an overall accuracy of 99.63% with FAR of 0.67% and FRR of 0.84%.

Keywords: Palmprint, Speech signal, Sum rule and Fusion.

1 Introduction

Biometrics refers to the use of physiological or biological characteristics to measure the identity of an individual. These features are unique to each individual and remain unaltered during a person's lifetime. These features make biometrics a promising solution to the society. The various biometrics traits available are speech signal, face, fingerprint, iris, palmprint, hand geometry, ear etc. Among the available biometric traits some of the traits outperform others. The reliability of several biometrics traits is measured with the help of experimental results. The biometric system is basically divided into two modes i.e., unimodal biometric system and multimodal biometric system. In case of unimodal biometric system the individual trait is used for recognition or identification. The system performs better under certain assumptions but fails

* Corresponding author.

when the biometric data available is noisy. The system also fails in case of unavailability of biometric template. Thus in such a situation multimodal biometric systems are used where more than one classifier is used to arrive at a final decision. The concept of multimodal biometric system has been proposed by Ross and Jain [1] where apart from fusion strategies various levels of integration are also presented. The score level fusion in multimodal biometrics system is proposed in [2]. A novel fusion at feature level for face and palmprint has been presented in [3].

In this paper a novel combination of two classifiers, i.e. palmprint and speech signal is presented. The problem can arise at the time of palmprint image acquisition. In case of speaker recognition poor quality of speech signal due to cold, throat infection may create problem. Thus the two recognizers are combined at matching score level and final decision about the person's identity is made. In the next section a brief system overview is presented about the palmprint and speech signal. Section 3 presents feature extraction using our own approach for palmprint and for speech signal, Sub-band based Cepstral Parameters and Gaussian Mixture Model is used. In Section 4 the two modalities are combined at matching score level. The experimental results prior to fusion and after fusion are presented in Section 5. Conclusion is given in the last section.

2 System Overview

The block diagram of a multimodal biometric system using two (palm and speech) modalities for human recognition system is shown in figure. It consists of three main blocks, that of Preprocessing, Feature extraction and Fusion. Preprocessing and feature extraction are performed in parallel for the two modalities. The preprocessing of the audio signal under noisy conditions includes signal enhancement, tracking environment and channel noise, feature estimation and smoothing [4].

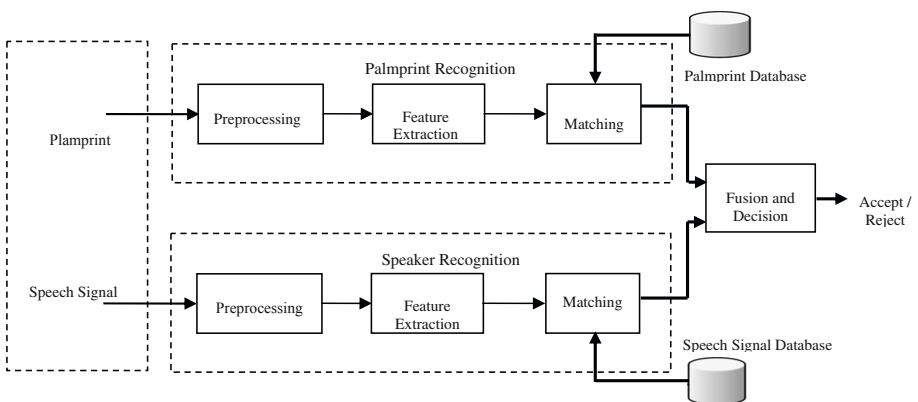


Fig. 1. Block diagram of the proposed multimodal biometric verification system

Further, features are extracted from the training and testing images and speech signal respectively, and then matched to find the similarity between two feature sets. The matching scores generated from the individual recognizers are passed to the decision module where a person is declared as genuine or an imposter.

3 Feature Extraction

3.1 Palmprint Feature Extraction Methodology

In the new approach we proposed, palmprint is identified by the total energy level in sectors of circular rings of the palm image. Details of the algorithm are as follows:

3.1.1 Identify Hand Image from Background

The setting of our system is such that we employ a contact-less capturing system that works without pegs. The background of the image is relatively uniform and is of a relatively low intensity when compared to the hand image. Using the statistical information of the background and that of the rest of the pixels, the algorithm estimates an adaptive threshold to segment the image of the hand from the background. Pixels with intensity above the threshold are considered to be part of the hand image.

3.1.2 Locate Region-of-Interest

Maximum palm area is extracted from the binary image of the hand. To ensure the extracted palm region has minimal rotation and translation error, the algorithm identifies gaps-between-fingers and uses them as reference points to align the image. By leveling the gap between the index finger (IF) and the middle finger (MF), and that between the MF and the ring finger (RF), images of the palm are aligned rotationally. Since the gap between MF and RF is usually the most healthy point amongst the three gaps, it is used as the reference point to eliminate translation Fig. 2(a).



Fig. 2. Figure (a) shows the of image alignment and figure (b) shows segmentation of ROI

The maximum square region that can fit in the selected palm area is chosen as the ROI. The square region is horizontally centered on the axis running through the gap between MF and RF. In addition, the ROI will be divided into non-overlapping circular-rings as shown in Fig. 2(b).

3.1.3 Feature Extraction

Preprocessing of palm images is necessary to minimize variations in palm images of the same individual. Firstly, a 2-D lowpass filter is applied to the image. Secondly, a Gaussian window is used to smooth out the image since Haar wavelet, due to its rectangular wave nature, is sensitive to noise.

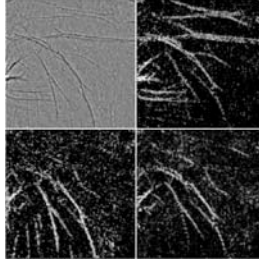


Fig. 3. Haar wavelet transform of Palmprint

A 1-level decomposition of the image by the Haar wavelet is carried out. For each of the three detail images obtained, i.e. image consisting of the horizontal, vertical and diagonal details, a smoothing mask is applied to remove noise. It was found that most of the low frequency components are attributable to the redness underneath the skin and should preferably be excluded from features for identification. Thus, pixels with frequency values within one standard deviation are set to zero. Values of the rest of the pixels are projected onto a logarithm scale so as to minimize the absolute differences in the magnitude of the frequency components between two images. That is,

$$I(x_i, y_i) = \begin{cases} 0, & \text{if } |I(x_i, y_i)| \leq \text{std}(I(x, y)) \\ \ln(|I(x_i, y_i)| - \text{std}(I(x, y)) + 1), & \text{o.w.} \end{cases} \quad (1)$$

where $I(x_i, y_i)$ is the frequency value in a detail image.

3.1.4 Feature Vector Construction

Each of the detail images is divided into non-overlapping circular-rings. The circles are all centered at the same point, with the area of them increasing by a factor of 4, i.e. the radius of a circle doubles that of its immediate inner one. Each circle is separated into a different number of sectors. The innermost circle is considered as 1 sector, while moving out from it, each outer circle will have 2 more sectors than its inner layer (see Fig. 4(a)).

Mean energy level of each sector, i.e. the total absolute sum of the frequency components divided by the number of pixels in each sector, is used to construct the feature vector (see Fig. 4(b)). The arrangement ensures that when two feature vectors of unequal length are compared to each other, point-wise comparison of them is actually comparing features in the same spatial region of the two different palm images.

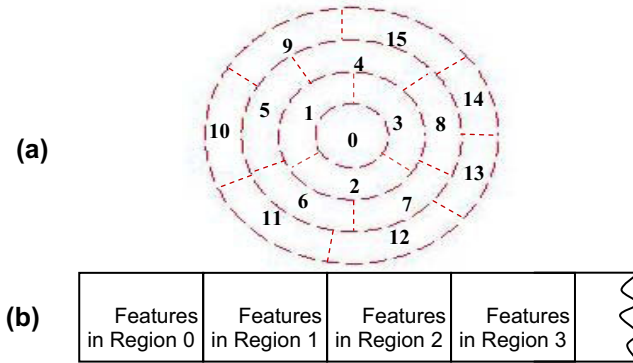


Fig. 4. Feature vector

3.1.5 Matching Score Calculation

Since the palm images under different palm sizes will result in feature vectors of different lengths. Due to the possibility of having variations in the extent the hand is stretched, the resultant maximum palm area may vary within the same subject. Therefore, the distance measure used must be able to fairly compare two feature vectors with unequal dimension. The score is calculated as the mean of the absolute difference between two feature vectors. If *featureV_i* represents a feature vector of *N_i* elements, the score between two images is given as:

$$Score(i, j) = \frac{\sum_{n=1}^{\min(N_i, N_j)} | featureV_i(n) - featureV_j(n) |}{\min(N_i, N_j)} \tag{2}$$

3.2 Subband Based Cepstral Parameters and Gaussian Mixture Model

3.2.1 Subband Decomposition via Wavelet Packets

A detailed discussion of wavelet analysis is beyond the scope of this paper and we therefore refer interested readers to a more complete discussion presented in [5]. In continuous time, the Wavelet Transform is defined as the inner product of a signal *x(t)* with a collection of wavelet functions $\psi_{ab}(t)$ in which the wavelet functions are scaled (by *a*) and translated (by *b*) versions of the prototype wavelet $\psi(t)$.

$$\psi_{a,b}(t) = \psi\left(\frac{t-b}{a}\right) \tag{3}$$

$$W_\psi x(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} x(t) \psi\left(\frac{t-b}{a}\right) dt \tag{4}$$

The wavelet transform which is obtained by iterating on the low pass branch, the filterbank tree can be iterated on either branch at any level, resulting in a tree

structured filterbank which we call a wavelet packet filterbank tree. The resultant transform creates a division of the frequency domain that represents the signal optimally with respect to the applied metric.

3.2.2 Wavelet Packet Transform Based Feature Extraction Procedure

Here, speech is sampled at 8 kHz. A frame size of 24msec with a 10msec skip rate is used to derive the Subband based Cepstral Parameters features. We have used the same configuration proposed in [6] for Mel Frequency Cepstral Coefficient (MFCC). Next, the speech frame is Hamming windowed and pre-emphasized.

The proposed tree assigns more subbands between low to mid frequencies while keeping roughly a log-like distribution of the subbands across frequency. The wavelet packet transform is computed for the given wavelet tree, which results in a sequence of subband signals or equivalently the wavelet packet transform coefficients, at the leaves of the Tree. In effect, each of these subband signals contains only restricted frequency information due to inherent bandpass filtering. The complete block diagram for computation of Subband based Cepstral Parameters is given in Fig. 5. The subband signal energies are computed for each frame as,

$$S_i = \frac{\sum_{m \in i} [(W_\psi)(i), m]}{N_i} \tag{5}$$

W_ψ : Wavelet packet transform of signal x , i :subband frequency index ($i=1,2...L$),

N_i : number of coefficients in the i^{th} subband.

3.2.3 Subband Based Cepstral Parameters

As in MFCCs the derivation of parameters is performed in two stages. The first stage is the computation filterbank energies and the second stage would be the decorrelation of the log filterbank energies with a DCT. The derivation of the Subband Based Cepstral parameters follows the same process except that the filterbank energies are derived using the wavelet packet transform. It will be shown that these features outperform MFCCs. We attribute this to the computation of subband signals with smooth filters. These parameters have been shown to be effective for speech recognition in car noise [7] and for classification of stressed speech. Subband Based Cepstral parameters are derived from subband energies by applying the Discrete Cosine Transformation:

$$SBC(n) = \sum_{i=1}^L \log S_i \cos\left(\frac{n(i - 0.5)}{L} \pi\right), n = 1, \dots, n' \tag{6}$$

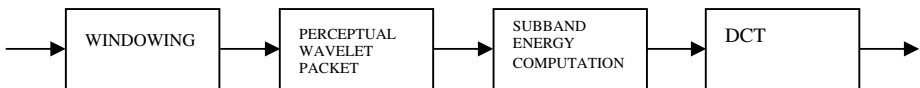


Fig. 5. Block diagram for Wavelet Packet Transform based feature extraction procedure

where n' is the number of SBC parameters and L is the total number of frequency bands. Because of the similarity to root-cepstral [8] analysis, they are termed as sub-band based cepstral parameters.

3.2.4 The Gaussian Mixture Model

In this study, a Gaussian Mixture Model approach proposed in [9] is used where speakers are modeled as a mixture of Gaussian densities. The Gaussian Mixture Model is a linear combination of M Gaussian mixture densities, and given by the equation,

$$p(\vec{x} | \lambda) = \sum_{i=1}^M p_i b_i(\vec{x}) \tag{7}$$

Where \vec{x} is a D -dimensional random vector, $b_i(\vec{x})$, $i=1, \dots, M$ are the component densities and p_i , $i=1, \dots, M$ are the mixture weights. Each component density is a D -dimensional Gaussian function of the form

$$b_i(\vec{x}) = \frac{1}{(2\pi)^{D/2} |\Sigma_i|^{1/2}} \exp \left\{ -\frac{1}{2} (\vec{x} - \vec{\mu})^T \Sigma_i^{-1} (\vec{x} - \vec{\mu}) \right\} \tag{8}$$

Where $\vec{\mu}$ denotes the mean vector and Σ_i denotes the covariance matrix. The mixture weights satisfy the law of total probability, $\sum_{i=1}^M p_i = 1$.

4 Fusion

The results generated from the individual traits are good but the problem arises when the user is not able to give speech signal correctly due to problem in throat and due to background noise. Thus in such a situation an individual cannot be recognized using the speech signals. Similarly, the problem faced by palmprint recognition system is the presence of scars and cuts. Thus to overcome the problems faced by individual traits of speech signal and palmprint, a novel combination is proposed for the recognition system. Scores generated from individual traits are combined at matching score level using weighted sum of score technique. Let MS_{Speech} and MS_{Palm} be the matching scores obtained from Speech signal and palmprint modalities respectively. The steps involved are:

4.1 Score Normalization

This step brings both matching scores between 0 and 1 [10]. The normalization of both the scores are done by

$$N_{Speech} = \frac{MS_{Speech} - \min_{Speech}}{\max_{Speech} - \min_{Speech}} \tag{9}$$

$$N_{palm} = \frac{MS_{palm} - \min_{palm}}{\max_{palm} - \min_{palm}} \quad (10)$$

where \min_{Speech} and \max_{Speech} are the minimum and maximum scores for speech signal recognition and $\min_{Palmprint}$ and $\max_{Palmprint}$ are the corresponding values obtained from palmprint trait.

4.2 Generation of Similarity Scores

The normalized score of palmprint which is obtained through Haar Wavelet gives the information of dissimilarity between the feature vectors of two given images while the normalized score from speech signal gives a similarity measure. So to fuse both the score, there is a need to make both scores as either similarity or dissimilarity measure. The normalized score of palmprint is converted to similarity measure by

$$N'_{palm} = 1 - N_{palm} \quad (11)$$

4.3 Fusion

The two normalized similarity scores are fused linearly using sum rule as

$$MS = \alpha * N'_{palm} + \beta * N_{Speech} \quad (12)$$

Where α and β are two weight values that can be determined using some function. In this paper a combination of linear and exponential function is used. The value of weight is assigned linearly if the value of matching score is less than the threshold. The value of MS is used as the matching score. So if MS is found to be more than the given threshold value the candidate is accepted otherwise it is rejected.

5 Experimental Results

The results are tested on speech signals and palmprint images collected by the authors. The database consists of six palm images and speech signals of each person with a total of 120 persons. The palm images are acquired using CCD camera with uniform light source. The speech signals are acquired using a microphone with uniform background noise. For the purpose of allowing comparisons, two levels of experiments are tested individually. At this level the individual results are computed and an accuracy curve is plotted as shown in Fig. 7. At this level the individual accuracy for palmprint and speech signal is found to be 92.79% and 94.64% respectively as shown in Table 1.

However in order to increase the accuracy of the biometric system as a whole the individual results are combined at matching score level. At second level of experiment the matching scores from the individual traits are combined and final accuracy graph is plotted as shown in Fig. 8. Table 1 shows the accuracy and error rates obtained from the individual and combined system. The overall performance of the

system has increased showing an accuracy of 99.63% with False Acceptance Rate (FAR) of 0.67% and False Rejection Rate (FRR) of 0.84% respectively. Receiver Operating Characteristic (ROC) curve is plotted for Genuine Acceptance Rate (GAR) against False Acceptance Rate (FAR) for individual recognizers and combined system as shown in Fig. 9.

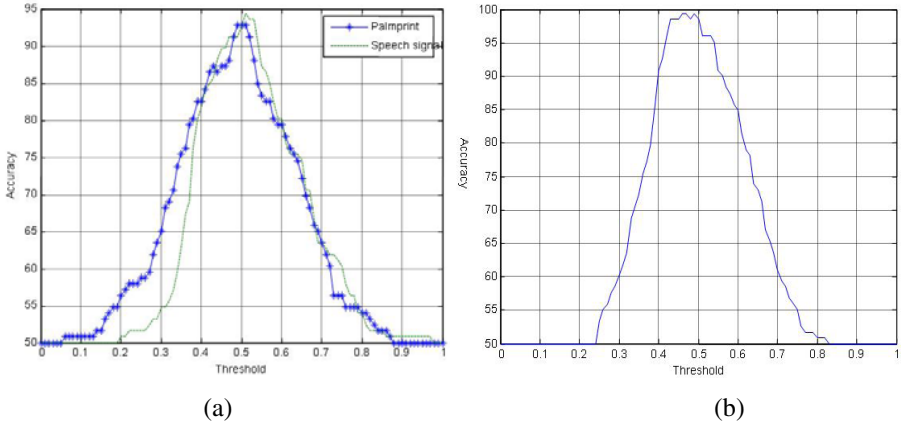


Fig. 6. Figure (a) shows accuracy plots of individual recognizers and figure(b) shows accuracy graph for combined classifier

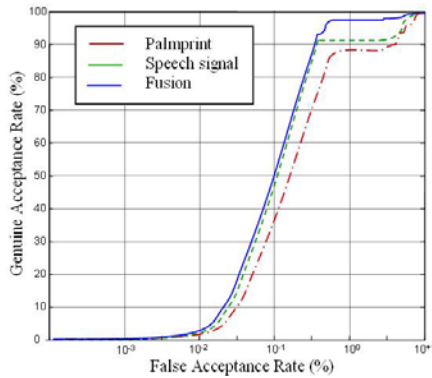


Fig. 7. ROC Curve for Palmprint, Speech signal and Fusion

Table 1. Figures showing individual and combined accuracy

Trait	Algorithm	Accuracy (%)	FAR (%)	FRR (%)
Palmprint	Haar Wavelet	92.79	2.72	4.73
Speech signal	SBC+GMM	94.64	5.87	1.35
Fusion	Haar + SBC	99.63	0.67	0.84

6 Conclusion

Biometric systems are widely used to overcome the traditional methods of authentication. But the unimodal biometric system fails in case of biometric data for particular trait. This paper proposes a new method in selecting and dividing the ROI for analysis of palmprint. The new method utilizes the maximum palm region of a person to attain feature extraction. More importantly, it can cope with slight variations, in terms of rotation, translation, and size difference, in images captured from the same person. Feature vectors are arranged such that point-wise comparison is matching features from the same spatial region of two different palms. Thus the individual score of two traits (speech & palmprint) are combined at classifier level and trait level to develop a multimodal biometric system. The performance table shows that multimodal system performs better as compared to unimodal biometrics with accuracy of more than 99%.

References

1. Jain, A., Bolle, R., Pankanti, S. (eds.): *Biometrics: Personal Identification in Networked Society*. Kluwer Academic, Boston (1999)
2. Jain, A.K., Ross, A., Prabhaker, S.: An Introduction to Biometric Recognition. In: *IEEE Trans. on Circuits and Systems for Video Technology, Special Issue on Image- and Video-Based Biometrics* (January 2004)
3. Zhang, D., et al.: Online Palmprint Identification. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 25(9), 1041–1050 (2003)
4. Rabiner, L., Juang, B.H.: *Fundamentals of speech Recognition*. Prentice-Hall, Englewood Cliffs (1993)
5. Rioul, O., Vetterli, M.: Wavelets and Signal Processing. *IEEE Signal Proc. Magazine* 8(4), 11–38 (1991)
6. Reynolds, D.A., Rose, R.C.: Robust Text_Independent Speaker Identification Using Gaussian Mixture Speaker Models. *IEEE Transactions on SAP* 3, 72–83 (1995)
7. Erzin, E., Cetin, A.E., Yardimci, Y.: Subband analysis for speech recognition in the presence of car noise. In: *ICASSP 1995*, vol. 1, pp. 417–420 (1995)
8. Alexandre, P., Lockwood, P.: Root cepstral analysis: A unified view: Application to speech processing in car noise environments. *Speech Communication* 12, 277–288 (1993)
9. Reynolds, D.A.: Experimental Evaluation of Features for Robust Speaker Identification. *IEEE Transactions on SAP* 2, 639–643 (1994)
10. Jain, A.K., Nandakumar, K., Ross, A.: Score Normalization in multimodal biometric systems. *The Journal of Pattern Recognition Society* 38(12), 2270–2285 (2005)

Logical Modeling and Verification of a Strength Based Multi-agent Argumentation Scheme Using NuSMV

Shravan Shetty, H.S. Shashi Kiran, Murali Babu Namala, and Sanjay Singh

Department of Information & Communication Technology,
Manipal Institute of Technology, Manipal University, Manipal-576104, India

Abstract. Software systems have evolved to the age of Artificial Intelligence (AI), consisting of independent autonomous agents interacting with each other in dynamic and unpredictable environments. In this kind of environment it is often very difficult to predict all the interactions between the agents. Hence verification of an interaction between multiple agents has become a key research area in AI. In this paper we model and verify an Automatic Meeting Scheduling (AMS) problem having multiple agent communication.

The AMS problem helps us to emulate a real life scenario where multiple agents can argue over the defined constraints. A *weighted strength based argumentation scheme* is proposed, where each argument is weighed against each other to determine the strongest evidence. The argumentation model described in the paper have six agents: Initiator, Scheduler and four Participant agents. We have formalized the agent interactions using Computation Tree Logic (CTL) and verified the scheme by providing suitable specifications (SPEC) in a symbolic model verifier tool called, NuSMV.

1 Introduction

There has been tremendous advances in the area of agent based argumentation systems. In such schemes, a common consensus is drawn from an interaction or argumentation between different entities called Agents. Currently agents can be considered as a component with independent behavior. Interactions and argumentation can be termed as agent communication. There is however a subtle difference between interaction and argumentation:

- Interaction can be a simple message passing scheme where the agents are exchanging knowledge with each other.
- Argumentation is a complex behavior where each of the agents will refute the other agent by providing its own evidence for a situation.

However both the communication types have a common result oriented approach; commonly termed as Desire in Belief, Desire and Intention (BDI Logic) [1]. This was originally developed so as to emulate human practical reasoning [3].

Agent knowledge can be classified as:

- Internal: Internal knowledge is specific to one agent and its view of reaching the overall system objectives. This is equivalent to Belief of the BDI logic [1].
- Global: Global knowledge is shared by all agents in the system. The overall system objectives is known by each of the agent.

Typically an argumentation takes place based on the individual internal (local) knowledge. In an attempt to model such an argumentation scheme, we have introduced a novel concept of *argument strength*, which will eventually decide the outcome of an argument. This argumentation not only makes the interaction efficient but also improves the outcome of the dialogue between the agents. Katie Atkinson [9] has given an argumentation scheme based on Command Dialogue Protocol (CDP). This scheme uses dialogue interactions between commander-agent and the receiver-agent. An argumentation in this scheme is based on the evidence that is provided. In our paper we employ a similar approach but rely on the strength of an argument for verification and simulation via model checking.

Similarly in [2] presented a dialogue framework that allows different agents with some expertise to inquire about the beliefs of others and also share the knowledge about the effect of actions. Two argumentation models discussed in it allows defeasible reasoning about what to believe and defeasible reasoning about what to do. In our approach, we do not consider that any of the agent know about the belief of others. Each agent has an independent view of the overall objective, so we have considered that each of the interaction in an argument, to have a strength associated with it. This enables us to determine the agent with the strongest argument. The evidence or the knowledge provided by this agent can be accepted.

We put forward a multi-agent modeling scheme using Argument Strength. We illustrate this argumentation scheme using a multi-agent meeting scheduling problem. We have formalized the agent interactions using Computation Tree Logic (CTL) [7] and verified the scheme by providing suitable specifications (SPEC) in a symbolic model verifier tool called, NuSMV [6]. The user manual for using this tool can be found in [5].

The remaining paper is organized as follows. Section 2 gives a theoretical background of agents and their interaction schemes. Section 3 gives the methodology employed in the AMS problem and its constraints. Section 4 expose us to a model checking environment used to verify the specifications identified. The verification of the model developed using NuSMV along with its trace results are mentioned in Section 5. Finally conclusion and areas of future work has been drawn in Section 6.

2 Theoretical Background

2.1 Agents

An Agent is a component capable of performing independent actions. It can be thought of as a simple software module or even as complicated as a robot with artificial intelligence. Every agent is associated with attribute, behavioral rules,

small memory resources, decision making sophistication and rules to modify behavioral rules [8]. An agent is autonomous and can function independently in its environment and in its interactions with other agents. Agents represent entities in a model and agent relationships represent entity interaction.

2.2 Agent Based Modeling

Agent Based Modeling (ABM) is synonymous to Individual Based Modeling (IBM) [8]. The Buyer-Seller model [4] serves as an example for explaining a simple argumentation scheme. The seller-agent tries to sell a car based on the features available e.g. air bag. Both buyer and seller-agent have a global knowledge about existence of an air bag and its use. The internal knowledge of the seller would be the research technicalities about the air bag performance, whereas the buyers internal knowledge would be limited to articles in newspapers. However we can consider the sale of the car as the common goal. The interaction between the two agents can be modeled as an argument over the quality and need of an airbag. An argument ends when both the agents are convinced by the evidence provided, leading to the goal or desire as in the BDI model [1]. An argument posed by each is weighted against each other in the *strength* based scheme. For instance, the strength of a seller-agent argument is more, as research data has empirical proof over its claim as compared to the newspaper as that of the buyer-agent.

2.3 Applications with Multi-Agent Communication

An argumentation in the real world is much complex as independent behaviors of each agent is included. Multi-Agent modeling is being used in wide application domains. One critical areas is the *Air Traffic Control* (ATC), where various agents communicate with each other with a common objective of safe and secure air traffic handling. They also evaluate the performance of an air traffic management system by interacting with each other.

3 Methodology

To understand an Agent interaction scheme, we have identified a Multi-Agent meeting scheduler problem [10]. It can be visualized as a real life meeting, where an agents such as initiator, scheduler and the various participants of the meeting will look forward to schedule a meeting based on individual constraints. For initiating the meeting the initiator-agent might pick prerequisites like the date, time, location and the resources needed. This is supplied to the scheduler without keeping in mind any other conflicting meeting. The conflict can be with the unavailability of any of the prerequisites of the meeting. Under such conflicts, an argumentation takes place between the agents. We propose a scheme for modeling such argumentation based on the *Strength of an Argument*.

Argumentation is relatively easy to model when one of the argument is true and the other is false. We can consider the outcome of an argument based on

the *true* argument. But when we consider a scheme where both the arguments are true, it is often difficult to model. Here we consider strength of an arguments for modeling the right outcome. In the Initiator-Scheduler argumentation, the strength of an argument by the scheduler-agent with respect to the meeting constraints are more as compared to the initiator-agent.

Table 1. Notations Used

Sl.No.	Notation	Entity
1	<i>I</i>	Initiator-agent
2	<i>S</i>	Scheduler-agent
3	<i>P</i>	Participant-agent
4	<i>I_argument</i>	Initiator-argument
5	<i>S_argument</i>	Scheduler-argument
6	<i>P_argument</i>	Participant-argument

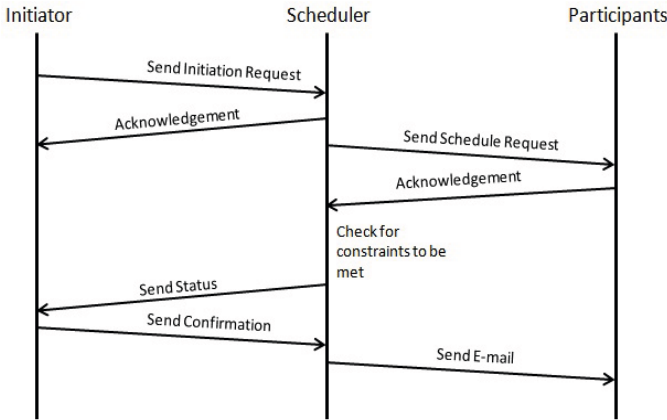


Fig. 1. Agent Interaction Diagram

The meeting scheduling problem [10] gives us a real life scenario where an agent based model can be used in order to schedule a meeting. An argumentation based on individual agent knowledge is identified and strength associated for each is considered for modeling. The Table 1 gives the various notations used henceforth, to identify each agent or their communication.

The interaction between the agents are as shown in Fig.1. *I* sends a request for the meeting by sending a message to *S*. *S* communicates between an *I* and *P*. It is responsible for establishing a meeting based on the responses of each participants. An argumentation is modeled over the meeting constraints such as date, time, location and availability of other resources (e.g. projectors, pointers etc). There will be an argument between any of the two agents whenever there is a conflict in the meeting constraints. For instance, the initiator-agent might pick a meeting date without looking into any conflicting constraints. The scheduler-agent checks the

date and passes an argument to refute the initial request. In this case, the strength of the S argument can be considered to be more than that of the I argument. An argument between the S and P are also similar. The strength of P _argument over the meeting constraints is more than that of I or S .

Algorithm 1. ScheduleMeeting(I, S, P)

```

1: Initiate(Date, Venue, Participant_List)
2: for each argument exchanged between  $I$  and  $S$  do
3:   Accepted $\leftarrow$ AssessStrength( $I$ _argument,  $S$ _argument)
4: end for
5: if  $I$ _argument is accepted then
6:   send Schedule request to all participants
7:   for each argument exchanged between  $S$  and  $P$  do
8:     Accepted $\leftarrow$ AssessStrength( $S$ _argument,  $P$ _argument)
9:   end for
10:  if  $S$ _argument is accepted then
11:    send meeting confirmation to the participants
12:  else
13:    cancel the meeting
14:  end if
15: else
16:   cancel the meeting
17: end if

```

The pseudo-code in Algorithm 1 explain the argumentation scheme. Line 1 is used to initiate the meeting with the constraints date, venue and the intended participant list. These details are sent to S , any argumentation hence formed is weighed against each other and the stronger argument is accepted as shown in Line 3. The argumentation scheme proceeds only if the I _argument is stronger, otherwise the meeting cannot be scheduled. The S _argument is modeled to be stronger in situation such as unavailable dates or meeting rooms (venue). Line 7 shows a possible argumentation between S and P , a corresponding acceptance in Line 8. If P _argument is stronger then the meeting is canceled. This situation can be considered when the participants are busy, the venue is far, etc.

Algorithm 2. AssessStrength($Argument1, Argument2$)

```

1: CalculateStrength( $Argument1, Argument2$ )
2: Accept Argument with higher score
3: return Accepted Argument

```

Algorithm 2 is used to assess any two arguments between agents by referring to Algorithm 3, which computes the strength of each argument. The argument having higher score will equivalently have a higher strength.

Algorithm 3. CalculateStrength(*Argument1*, *Argument2*)

- 1: Score1 := *Argument1*.agent_weight + $\sum_{i=1}^N$ *ConstraintWeight_i*
- 2: Score2 := *Argument2*.agent_weight + $\sum_{i=1}^N$ *ConstraintWeight_i*
/* where N : is the number of constraints */
- 3: **return** max(Score1, Score2)

4 Model Checking

Over the years, model checking has evolved greatly into the software domain rather than being confined to hardware such as electronic circuits. Model checking is one of the most successful approach to verification of any model against formally expressed requirements. It is a technique used for verifying finite state transition system. The specification of system model can be formalized in temporal logic, which can be used to verify if a specification holds true in the model.

```

C:\Windows\system32\cmd.exe - nusmv -int "KnowledgeAgentMeeting v 2.smv"
*** This is NuSMV 2.4.3 (compiled on Tue May 22 14:08:54 UTC 2007)
*** For more information on NuSMV see <http://nusmv.first.itc.it>
*** or email to <nusmv-users@first.itc.it>.
*** Please report bugs to <nusmv@first.itc.it>.

*** This version of NuSMV is linked to the MiniSat SAT solver.
*** See http://www.cs.chalmers.se/Cs/Research/FormalMethods/MiniSat
*** Copyright (c) 2003-2005, Niklas Fen, Niklas Sorensson

NuSMV > go
NuSMV > check_ctlspec
- specification (!i.initiate_request -> AF status) is false
-- as demonstrated by the following execution sequence
Trace Description: CTL Counterexample
Trace Type: Counterexample
-> State: 1.1 <-
  status = 0
  flexible = 0
  i.initiate_request = 0
  i.i_status = 0
  i.i_support.room = 1
  i.i_support.dates = 1
  i.i_support.resource = 1
  i.i_support.strength = 0
  s.schedule_request = 0
  s.s_status = 0
  s.s_ack = 0
  s.s_support.room = 1
  s.s_support.dates = 1
  s.s_support.resource = 1
  s.s_support.strength = 0
  p.active = 0
  p.important = 0
  p.audience1 = 0
  p.audience2 = 0
  p.ready = 0
  p.p_ack = 0
  p.p_support.room = 1
  p.p_support.dates = 1
  p.p_support.resource = 1
  p.p_support.strength = 0
  l.lessers = 1
  i.i_support.CurrentStrength = 4
  s.s_support.CurrentStrength = 5
  p.p_support.CurrentStrength = 6
-> Input: 1.2 <-
-- Loop starts here
-> State: 1.2 <-
  i.i_support.strength = 4
  s.s_support.strength = 5
  p.p_support.strength = 6
-> Input: 1.3 <-
-> State: 1.3 <-
-- specification (AG i.initiate_request -> EF !s.schedule_request) is true
NuSMV >

```

Fig. 2. Screen shot depicting the NuSMV implementation of CTL SPEC No. 1 and 2

Model checking has a number of advantages over traditional approaches which are based on simulation, testing and deductive reasoning. In particular, model checking is an automatic, fast tool to verify the specification against the model. If any specification is *false*, model checker will produce a counter-example that can be used to trace the source of the error. We have modeled the multi-agent meeting schedule and verified against few specifications in NuSMV. We have considered weights for each agent, which randomly picks a value from a range. The strength of an argument can then be assessed based on the higher score considering the meeting constraints. A simple method would be to assign range values to each constraint and a weight for each agent. The overall argument strength can be modeled as a score equal to the sum of the agent weight and the sum of the constraint weights. We have considered higher weight for the participant-agents.

5 NuSMV Simulation Results

5.1 Specifications

Like in most object oriented programming languages, NuSMV also uses `.` (dot) operator to access state variables of a module instance. For example in our implementation, *i* is an instance of the module *Initiator*. The state variables of the module *Initiator* can be accessed using *i*.*<state variables>*. Hence *i.i_support.strength* implies the state variable *strength* of the instance *i_support* which in turn is a state variable of the instance *i*.

We have verified the multi-agent scheme by many suitable specifications, few of them has been listed in Table 2. Each of the specification is explained below:

- AG *i.initiate_request* \rightarrow EF status
This checks the property that when a meeting is initiated, does there exist a path such that the meeting is fixed? This returns *TRUE*.
- AG(!status)
This specification is used to simulate a trace of all the states for which the meeting gets fixed. The property can be interpreted as, is it always the case that the meeting is not fixed? This specification is *FALSE* and a counter-example simulation is generated by NuSMV showing the state transitions needed for fixing the meeting.
- !*i.initiate_request* \rightarrow EF status
This specification verifies the property, whenever a meeting is not initiated, is there any path that will fix the meeting? This is *FALSE* and hence generates a counter-example but in no path the status of the meeting is set to true when the meeting is actually not initialized.
- AG *i.initiate_request* \rightarrow EF !schedule
The last specification verifies if there can be any path where the meeting is initiated but the scheduler does not forward this request to the participants. This is *TRUE*, as a request by the initiator can be canceled by the scheduler directly when the meeting constraints are not met (e.g. date or venue etc).

Table 2. Specifications for the Meeting schedule

Sl.No.	Specification	Satisfiability
1	$(i.initiate_request \rightarrow EF \text{ status})$	False(Counter-example)
2	$(AG \ i.initiate_request \rightarrow EF \ !schedule)$	True
3	$(AG \ i.initiate_request \rightarrow EF \ \text{status})$	True
4	$AG(\!status)$	False(Counter-example)

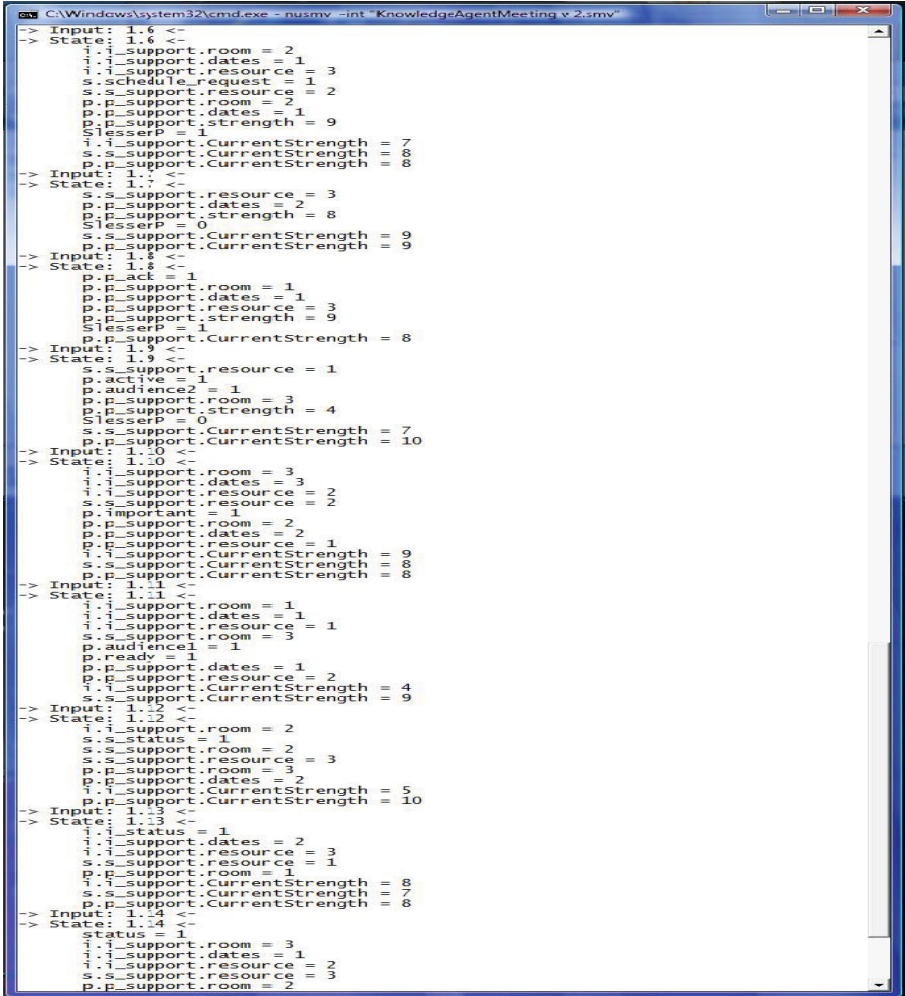


Fig. 3. Partial Trace Result

5.2 Trace Results

Trace in NuSMV demonstrates the transitions in state variables without the need of any specifications mentioned in the model. Each trace is identified by

a trace number and state numbers corresponding to each state. Each state is hence represented by `State:<TraceNumber.StateNumber>`. For example, `State:1.4` refers to the fourth state of the first generated trace. In every state only those variables having an updated value as compared to the previous state, are reflected and the rest are hidden.

In addition to the simulation results to verify the specification, we have also generated a trace result of the AMS problem discussed. Trace can be generated by using the `show_traces` command in NuSMV after the `.smv` code has been built. Figure 3 shows the trace result from states 1.6 to 1.14.

In state 1.6, `s.s.support.strength` and `p.p.support.strength` have values 8 and 9 respectively. As the strength of the P is higher, `p_ack` is not set to 1 in the next state 1.7. This is equivalent to a scenario where there is an argument between S and P with the latter having a stronger argument. But in state 1.7, both `s.s.support.strength` and `p.p.support.strength` have values 8. Hence in the next state 1.8, `p_ack` is set to 1. This shows a typical argumentation scheme in which each agent argues twice. In the first argument, P poses a stronger argument due to which an acknowledgment (i.e. `p.p_ack`) is not sent. This leads to a second argument where S refutes with a stronger argument, eventually leading to an acknowledgment from P .

6 Conclusion and Future Work

In this paper we have used an argumentation scheme based on *strength* for a multi-agent scenario. The meeting scheduler problem gives us a simple yet convincing model for designing an argumentation scheme. It also shows and explains the steps to identify the agents and their communication scheme. The *strength* based scheme is effectively used in modeling this communication scheme. It can be straightaway extended to a more complex scenario having many agents. We have also seen the formalization of the model and its corresponding specifications been verified and simulated using NuSMV. It is also easy to check the counter-examples generated by NuSMV simulation, so any modifications in the implementation can be easily done.

As future enhancements we would like to model *strength* of argument dynamically by incremental development of their knowledge.

References

1. Rao, A.S., Georgeff, M.: Bdi agents: From theory to practice. In: First International Conference on Multiagent System, pp. 312–319. AAAI Press, Stanford (1995)
2. Black, E., Atkinson, K.: Dialogues that account for different perspectives in collaborative argumentation. In: Decker, Sichman, Sierra, Castelfranchi (eds.) 8th Int. Conf. on Autonomous Agents and Multiagent Systems, AAMAS, Budapest, Hungary (2009)
3. Bratman, M.E.: Intentions, Plans, and Practical Reason. CSLI Publication, Stanford (1987), <http://csli-publications.stanford.edu/site/1575861925.shtml>

4. Caminada, M., Prakken, H.: Argumentation in agent system part 2: Dialogue. In: *Argumentation in Agent System*, Universiteit of Utrecht, Netherlands (2007)
5. Cavada, R., Cimatti, A., et al.: Nusmv 2.4 user manual (2009), <http://nusmv.irst.itc.it>
6. Cimatti, A., Clarke, E., et al.: NuSMV 2: An OpenSource Tool for Symbolic Model Checking. In: Brinksma, E., Larsen, K.G. (eds.) *CAV 2002*. LNCS, vol. 2404, pp. 359–364. Springer, Heidelberg (2002)
7. Huth, M., Ryan, M.: *Logic in Computer Science: Modelling and Reasoning about Systems*. Cambridge University Press, Cambridge (2005)
8. Macal, C., North, M.: Agent-based modeling and simulation. In: *Proceedings of the 2009 Winter Simulation Conference*, pp. 86–98. IEEE, Los Alamitos (2009)
9. Medellin, R., Atkinson, K., McBurney, P.: Model checking command dialogues. In: *Association for the Advancement of Artificial Intelligence*, pp. 58–63. AAAI Press, Stanford (2009)
10. Rolland, C., et al.: A proposal for a scenario classification framework. *Requirement Engineering* 3, 23–47 (1998)

Key Frame Detection Based Semantic Event Detection and Classification Using Heirarchical Approach for Cricket Sport Video Indexing

Mahesh M. Goyani¹, Shreyash K. Dutta², and Payal Raj³

^{1,2} Department of Computer Engineering, GCET, SP University, Anand, Gujarat, India

² Student Member IEEE, IEEE CS

³ Computer Engineering Department, SVMIT, VNSGU, Bharuch, Gujarat, India

mgoyani@gmail.com, shreyashdutta@gmail.com,

payal_mahida@yahoo.co.in

Abstract. In this paper, we propose a key frame detection based approach towards semantic event detection and classification in cricket videos. The proposed scheme performs a top-down event detection and classification using hierarchical tree. At level 1, we extract key frames for indexing based upon the Hue Histogram difference. At level 2, we detect logo transitions and classify the frames as realtime or replay fragments. At level 3, we classify the realtime frames as field view, pitch view or non field view based on colour features such as soil colour for pitch view and grass colour for field view. At level 4, we detect close up and crowd frames based upon edge detection. At level 5a, we classify the close up frames into player of team A, player of team B and umpire based upon skin colour and corresponding jersey colour. At level 5b, we classify the crowd frames into spectators, player's gathering of team A or player's gathering of team B. Our classifiers show excellent results with correct detection and classification with reduced processing time.

Keywords: Histogram, Template matching, Dominant Grass Pixel Ratio, Dominant Soil Pixel Ratio, Connected Component Analysis.

1 Introduction

In recent years sports video analysis has become a widely research area of digital video processing because of its huge viewership and commercial importance [1], [2]. The database of sports video is ever increasing, thus making information retrieval difficult and cumbersome. The quantity of data produced by recording sports videos need filtration and summarization. Video summarization and highlight extraction application satiate the need of sports fans to view only the exciting parts of broadcasts instead of wasting their time to watch in entirety. Genre-specific and genre-independent approaches are widely used among researchers for sports video analysis. Because of the enormous difference in sports videos, sport specific methods show successful results and thus constitute the majority of work. Some of the genre specific research have been done in soccer (football) [3], [4] tennis [5], cricket [6], basketball [7], volleyball [8], etc. less work is observed for genre-independent studies [8], [9].

Cricket is second to soccer when it comes to viewership and fan-following. Major cricket playing nations are India, Australia, Pakistan, South Africa, Zimbabwe, Bangladesh, Sri Lanka, New Zealand, England. Though cricket is spreading its roots to many other countries such as Kenya, Afghanistan, Ireland etc. In spite of its huge viewership cricket has not obtained its share in the research community [6], [10]. Cricket video analysis is far more challenging because of the complexities of game in itself. Cricket is a game of variable factors as compared to other famous sports such as soccer, basketball, tennis etc. Some of these are: 1) Format: cricket is played in three formats tests, one dayers and the new popular addition -T20. 2) No standard playing Conditions: the field area, pitches etc. vary from venue to venue. 3) Day and day/night matches cause illumination related problem. 4) Duration: cricket is one of the longest duration games thus generating huge amounts of data and enormous computation time. Even the shortest version of cricket i.e.T20 is played for approximately 3 hours which is greater than soccer (approximately 90 minutes) and hockey (approximately 70 minutes).

We propose a key frame detection approach for minimizing the computation time which is imperative as the amount of data is huge. We have concentrated our work on T20 matches as they offer lots of events in short duration as compared to ODI's. The same work can be extended to ODIs as well.

The paper is further organized as follows. Section 2 shows the classification and detection algorithms. Section 3 the experimental results and Section 4 concludes the paper with related future work.

2 Detection and Classification

2.1 Level 1: Key Frame Detection

We have observed that the cricket frame sequences have certain properties like some frame sequences are either same or have a little difference between them as shown in the fig. 2, thus using classifiers to classify all such frames is computationally inefficient. Suppose a sequence of same(S) and different (D) frames be:

S S S D S S S

For such a sequence of frames there would be a spike in the HHD plot with a rising edge at frame 4 and falling edge at frame 5, we have to consider only the rising edges to decide on key and non key frames. As shown in Fig. 2 Frame #580 is classified as a key frame.

Algorithm 1:

1. Convert the input RGB into HSV image format.
2. Compute the frame – frame hue histogram difference by using the formula:

$$Diff(i) = \frac{1}{r * c} \sum_{n=1}^{256} |hist[i](n) - hist[i-1](n)| \quad (1)$$

3. **if** (Diff(i)-Diff(i-1) > kfthresh) **then**
Classify the frame i as key frame.

else

Non key frame no indexing required.

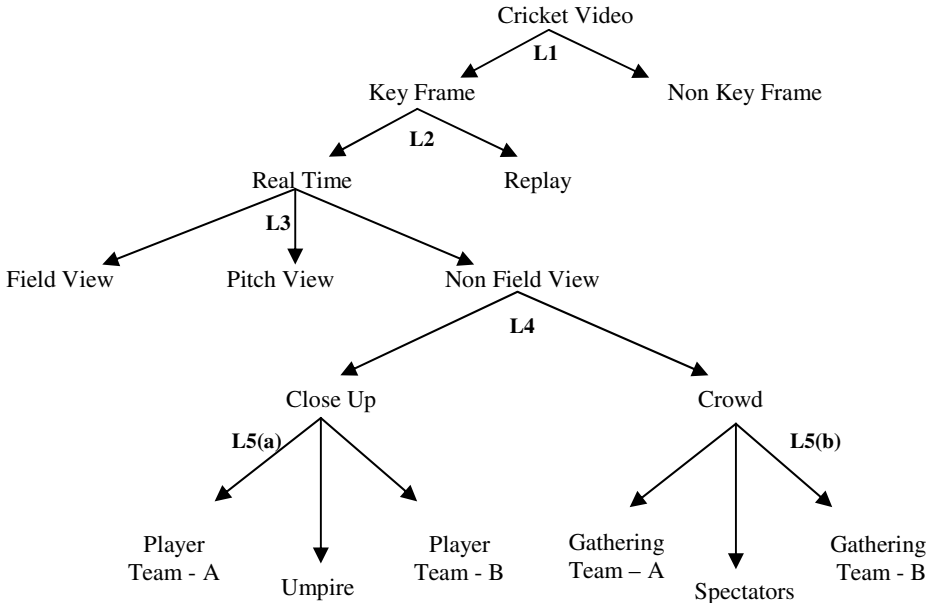


Fig. 1. Hierarchical framework followed for detection and classification

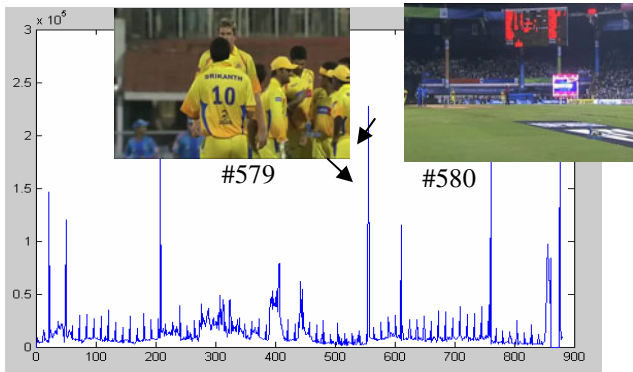


Fig. 2. Hue histogram difference graph for key frame detection

2.2 Level 2: Replay /Realtime Segments Detection

To detect replays from sports video, some early works focus on the characteristics of them such as motion vector [6] and replay structures [11]. However, these methods are not robust enough to be suitable for various kinds of sports video replay detection because replays in different sports video are various and compiled in different manners and can hardly be represented by such simple features. Therefore the recent approach is to detect the accompanying logo effect of the replays in sports videos to acquire the replay segmentations. [12, 13]. A replay segment is always sandwiched

between two logo transitions or flying graphics which last for 8-15 frames. We have used the logo colour property to detect logo frames or flying graphics and thus detect replay and live segments.

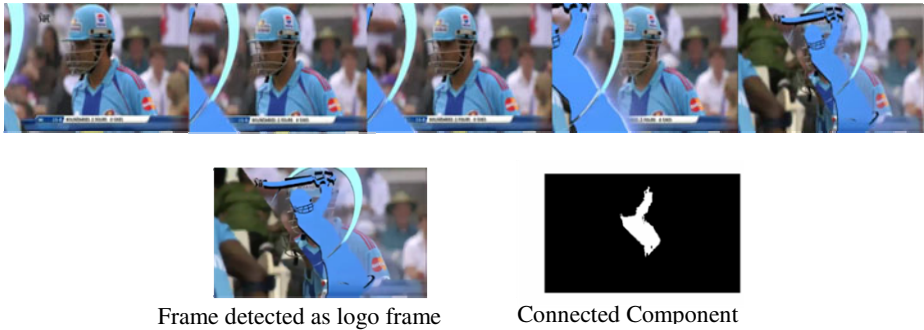


Fig. 3a. Example of a logo transition and detection

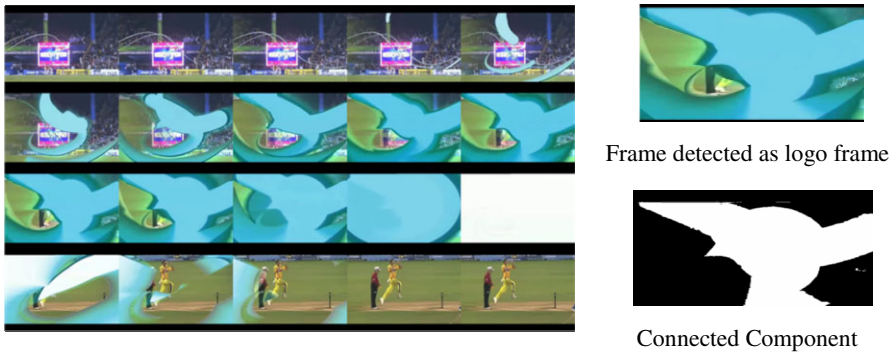


Fig. 3b. Example of a flying graphics

Algorithm 2:

1. Convert the input RGB frame into HSV image format.
2. Compute 256 bin hue and saturation histogram of the frame.
3. Use the condition for detecting logo pixels
if $(t1 < s < t12) \wedge (t3 < h < t4)$ **then**
 pixel belongs to logo color.
- else**
 pixel does not belong to logo color.
4. Apply connected component technique to remove noisy logo detected object from the images.
5. Compute the number of logo pixels - lnum .
if $lnum > lthresh$ **then**
 the frame belongs to class logo.
 skip t5 number of frames to avoid multiple logo frame detections.
- else**
 the frame doesn't belongs to class logo

2.3 Level 3: Pitch View/ Field View Detection

Pitch view and Field view can be detected using the color properties.

2.3.1 Pitch View

A pitch in a cricket field can be easily distinguished by its color, using this property we calculate the DSPR (i.e. dominant soil pixel ratio) to decide on the pitch view and non pitch view. We plot 256-bin histogram of the hue component of the frames. We pick up the peaks of hue histogram of these images. As shown in Figure 4, we observed peak at bin $k = 23$ and value of the peak is 15100 for the particular image of size 270×520 . By testing 50 images, we observed that the soil color peak occurs between bin $k = 20$ to $k = 25$. The summation of peaks of the histogram gives number of the pixels of the soil in the image (xs). From this, we compute the DSPR as xs/x , where x is the total number of pixels in the frame. We observed DSPR values vary from 0.05 to 0.1 for the pitch view. For non-pitch view image shown in Figure 6, we observed peaks belongs to the bins other than $k = 20$ to $k = 25$ and DSPR value is very small. As the soil color of pitches in a cricket field change from venue to venue change in bins is also possible so it is important to compute the histogram first then observe the bins.

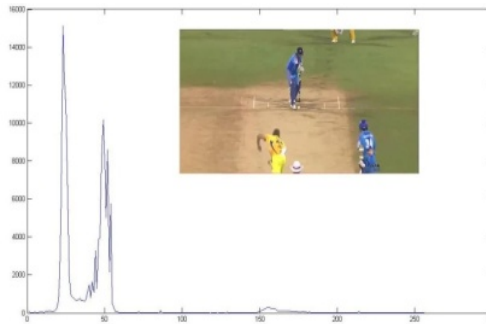


Fig. 4. Hue histogram of a Pitch view

Algorithm 3 :

1. Convert the input *RGB* frame into *HSV* image format and plot 256-bins Hue-histogram.
2. Compute DSPR by observing the peaks between bins $k = 20$ to $k = 25$.
3. The image can be classified as field view or non-field View by using following condition:
if (DSPR >sthresh) **then**
 frame belongs to class pitch view
else
 frame belongs to class non-pitch view

2.3.2 Field View

A field view image can be distinguished by its grass color as in, using this property we calculate the DGPR[14](i.e. dominant grass pixel ratio) to decide on the field view and non field view images. We plot 256-bin histogram of the hue component of

these images. We pick up the peaks of hue histogram of these images. As shown in Figure 5, we observed peak at bin $k = 56$ and value of the peak is 5090 for the particular image of size 270×520 . By testing 50 images, we observed that the green color peak occurs between bin $k = 48$ to $k = 58$. The peak of the histogram gives number of the pixels of the grass in the image. We call this number as x_g . From this, we compute the dominant grass pixel ratio ($DGPR$) as x_g/x , where x is the total number of pixels in the frame. We observed $DGPR$ values vary from 0.25 to 0.5 for the field view. For non-field view image shown in Figure 6, we observed peak belongs to the bins other than $k = 48$ to $k = 58$ and $DGPR$ value is very small. As the grass color of pitches in a cricket field change from venue to venue change in bins is also possible so it is important to compute the histogram first then observe the bins.

Algorithm 4 :

1. Convert the input *RGB* frame into *HSV* image format and plot 256-bins Hue-histogram.
2. Compute $DSPR$ by observing the peaks between bins $k = 48$ to $k = 58$.
3. The image can be classified as field view or non-field view by using following condition:
if ($DGPR > 0.25$) **then**
 frame belongs to class field view.
else
 frame belongs to class non field view.

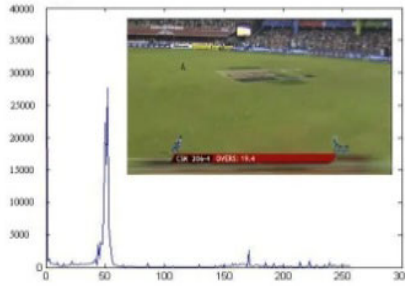


Fig. 5. Hue histogram of a field view image

2.4 Level 4b: Close Up / Crowd Detection

At this level we try to detect close up and crowd frames from the non pitch/non field view frames detected at level 3. From our observations we have seen that close up or crowd frames are shown frequently whenever an exciting event occurs such as when a wicket falls close up of batsman and bowler, then view of spectators and the player’s gathering of fielding team are certainly shown. Thus detecting such frames is imperative for concept mining.



Fig. 6. (a) Close up image. (b) Edge detection results of fig.(a). (c) Crowd image. (d) Edge detection results of fig. (c).

We have used edge detection feature to distinguish between close up and crowd frames, close up frames as shown in Fig. 6(b) have lesser edge pixel density then the crowd frames shown in Fig. 6(d).

Algorithm 5:

1. Convert the input RGB frame into YCbCr image format.
2. Apply *canny* edge detection operator to detect the edge pixels.
3. Count the number of edge pixels (EP) in the image.
4. Use the following condition for classification:
 - If** (EP > EPthresh) **then**
frame belongs to class crowd.
 - else**
frame belongs to class close-up.

2.5 Level 5a: Close Up Classification

In this level, we will be classifying the close up frames by using the skin color and jersey color information. Because of race, face geometry, helmet, caps etc. factors the face detection is not an easy task [15]. Fig. 8 shows some of the close up frames extracted from cricket videos. It can be observed that skin blocks are mostly detected in the blocks 6, 7, 10, 11. According to the face block location we will select the corresponding jersey block for close up classification [14]. Algorithm 6 details our approach.



Fig. 7. Some representative close up (a), (b) Close up of batsman. (c) Close up of Umpire.

Table 1. Skin block/s to jersey block/s correspondence table

Index	Skin block/s detected	Corresponding jersey block/s
1	S6,S7,S10,S11	8,12
2	S6,S7	8
3	S10,S11	12
4	S6,S10	7,11
5	S7,S11	8,12
6	S6	7
7	S7	8
8	S10	11
9	S11	12

Algorithm 6:

1. Convert the input image into YC_bC_r image format for skin pixels detection.
2. Use the following condition for detecting skin pixels.
If $(60 < Y < 120)$ and $(100 < C_b < 130)$ and $(C_r > 135)$ **then**
pixel belongs to skin color
else
pixel does not belong to skin color
3. Divide the image into 16 blocks as shown in figure 9, and compute the percentage of skin color pixels in each Block.
4. Let S_6, S_7, S_{10}, S_{11} are the skin percentages of block number 6, 7, 10, 11 respectively. Select threshold *skinthresh* for considering the block as skin block.
5. Check the skin block to jersey block correspondence table (Table 1).
6. Compute the 256 bins Hue-Histogram and Saturation-Histogram of the corresponding jersey block.
7. Classify the frames using the below pseudo code :
For each class i
do compute the ratio of number of jersey color pixels to total number of pixels- $djpr$
if $djpr > cthresh$ **then**
Classify the frame as close up class i
end
end

2.6 Level 5b: Crowd Classification

In this level we try to classify the crowd frames detected at level 4 using the jersey color information into the following three classes: Players Gathering of team A, Players gathering of team B and spectators. Since fielders gather on the field after exciting events such as wicket fall and third umpire referrals classifying them is important. Spectators are also shown at all important junctures.



Fig. 8. (a), (b): Player's gathering of teams. (c):Spectators. (d):Jersey color detection of Fig. 8(b).

Algorithm 7:

1. Convert the input image into HSV color space.
2. Compute the 256 bins Hue-Histogram and Saturation – Histogram of the corresponding jersey block.
3. Classify the frames using the below algorithm:
For each team class i
do compute the ratio of number of jersey color pixels to the total number of pixels – $djpr$.
If $djpr > crthresh$ **then**

```

        classify frame as players crowd of class i.
    end
end
if class not found then
    classify the frame as spectators.

```

3 Results and Conclusions

For our experiments we have used videos of IPL T20 matches as T20 matches generate a lot of events in a short span of time, which are enlisted in Table 2. Table 3 shows the number of frames we have employed for the processing to reduce the processing time. Table 4 shows number of various kind of key frames, detected for all three video segments at various level.

Table 2. Experimental Database

Cricket Video	Cup	Match(A vs. B)
V1	IPL 2008	Match 8 : CSK vs. MI
V2	IPL 2008	Match 36: MI vs. CSK.
V3	2007 World T20	Final match : India vs. Pakistan

Table 3. Sample Segments

Segment	Video	Total Frame	Time
1	V1	657	26.28
2	V1	882	35.28
3	V2	796	31.84

Table 4. Statistics of frame detection and classification

Features	Segment -1	Segment -2	Segment-3
Key Frame	56	104	108
NonKey Frame	601	778	688
Replay	108	21	325
Realtime	549	861	471
Pitch View	154	28	95
Field View	0	179	150
Close up	30	46	332
Crowd	364	506	254

Our Key Frame detection based approach shows excellent detection accuracy and also results in saving of processing time. The classifiers exhibit better detection and classification ratio at various levels. Our algorithms are able to extract out events as shown in table 4 from cricket videos. Snapshots in fig. 9 show some of the classified frames. Caption in top of the frame shows the detected event for that particular frame.



Fig. 9. Some of the classified frames

References

1. Kokaram, A., Rea, N., Dahyot, R., Tekalp, M., Bouthemey, P., Gros, P., Sezan, I.: Browsing sports video: trends in sports-related indexing and retrieval work. *IEEE Signal Processing Magazine* 23(2), 47–58 (2006)
2. Li, Y., Smith, J., Zhang, T., Chang, S.: *Multimedia Database Management Systems*. Elsevier *Journal of Visual Communication and Image Representation*, 261–264 (2004)
3. Wang, J., Chng, E., Xu, C., Lu, H., Tian, Q.: Generation of personalized music sports video using multimodal cues. *IEEE Transaction on Multimedia* 9(3) (2007)
4. Baillie, M., Jose, J.M.: Audio-based event detection for sports video. In: Bakker, E.M., Lew, M., Huang, T.S., Sebe, N., Zhou, X.S. (eds.) *CIVR 2003*. LNCS, vol. 2728, Springer, Heidelberg (2003)
5. Zhu, G., Huang, Q., Xu, C., Xing, L., Gao, W., Yao, H.: Human behavior analysis for highlight ranking in broadcast racket sports video. *IEEE Transactions on Multimedia* 9(6) (2007)
6. Kolekar, M.H., Sengupta, S.: Event-importance based customized and automatic cricket highlight generation. In: *IEEE Int. Conf. on Multimedia and expo* (2006)
7. Xu, C., Wang, J., Lu, H., Zhang, Y.: A novel framework for semantic annotation and personalized retrieval of sports video. *IEEE Transactions on Multimedia* 10(3) (2008)
8. Duan, L., Xu, M., Tian, Q., Xu, C., Jin, J.: A unified framework for semantic shot classification in sports video. *IEEE Transactions on Multimedia* 7(6) (2005)
9. Hanjalic, A.: Generic approach to highlight extraction from a sport video. In: *ICIP* (2003)
10. Xu, P., Xie, L., Chang, S., Divakaran, A., Vetro, A., Sun, H.: Algorithms and system for segmentation and structure analysis in soccer video. In: *IEEE ICME* (2001)
11. Pan, H., Beek, P., Sezan, M.: Detection of slow-motion replay segments in sports video for highlights generation. In: *ICASSP 2001*, Salt Lake City, UT (May 2001)
12. Tong, X., Lu, H.: Replay detection in broadcasting sports video. In: *Proceedings of the Third International Conference on Image and Graphics* (2004)
13. Pan, H., Li, B., Sezan, M.I.: Automatic detection of replay segments in broadcast sports programs by detection of logos in scene transitions. In: *Proc. IEEE, ICASSP* (2002)
14. Kolekar, M.H., Palaniappan, K., Sengupta, S.: Semantic event detection and classification in cricket video sequence. In: *Sixth Indian Conference on Computer Vision, Graphics & Image Processing*
15. Goyani, M., Joshi, B., Shikkenawis, G.: Acceptance/Rejection Rule Based Algorithm For Multiple Face Detection in Colour Images. *International Journal of Engineering Science And Technology* 2(6), 2148–2154 (2010)

Towards a Software Component Quality Model

Nitin Upadhyay¹, Bharat M. Despande¹, and Vishnu P. Agrawal²

¹ Computer Science & Information Systems Group, BITS-Pilani Goa Campus,
NH-17 B Zuari Nagar,
Goa, Goa-403726, India
{nitinu,bmd}@bits-goa.ac.in

² Mechanical Engineering Department, Thapar University, Research & Development Campus,
Patiala, Punjab-403726, India
vpagrawal@thapar.edu

Abstract. The academic and commercial sectors have noticed the growing impact of the component based software development (*CBSD*), in particular, to develop customizable, cost effective, just-in-time and reusable large scale and complex software systems. To build complete software solution the main focus is to create high quality parts and later join them together. One of the most critical processes in *CBSD* is the selection of software component as per end user criteria. Quality model plays an important role in the component selection process. This paper presents a software component quality model (*SCQM*) by overcoming shortcomings of existing quality models. Based upon this end user can take decision upon selection, evaluation and ranking of potential component candidates and wherever possible attain improvements in the component design and development.

Keywords: component, quality, component quality model, component quality characteristics.

1 Introduction

During the last years a significant change is noticed in the paradigm of software development. Modern software's are large scale and complex. Many organizations consider implementing such software using commercial off-the-shelf (*COTS*) components with the expectations that *COTS* components can significantly lower development costs, shorten development life cycle, produce high reliable and high stable product [1]. Component based software engineering has emerged as a separate discipline which ultimately leads to software system that require less time to specify, design, test and maintain and yet establishes high reliability requirements [2-5]. The quality of a component based software system depends upon the quality of individual components and interactions among them considering any rationale or constraint for selecting them [6-7]. A software quality model acts as a framework for the evaluation

of characteristics of software. It is to be noted that with the development of technology and research, software being used in an increasingly wide variety of application areas and its correct operation is often critical for business success [8-9]. To ensure adequate quality of a product it is necessary to perform comprehensive specification and evaluation of software component quality. This can be achieved by defining appropriate quality model/standard for the component market that will guarantee component acquirer, high quality components [10-11]. Existing quality models are inappropriate to evaluate quality of software components as they are very general in nature. Very few research studies which are available have contributed to the development of component specific quality model. More or less each and every model is the customization of ISO-9126 software quality model.

In this paper, we intend to review existing quality models (general and component specific) in order to propose comprehensive component specific quality mode to evaluate quality of a software component. The rest of the paper is structured as follows: In section 2, the existing quality models are reviewed. Section 3 describes the proposed component specific quality model which overcomes the shortcomings of existing quality models. Finally, conclusion is given which details out contribution made and future directions.

2 Review of Quality Models

The state of art literature does not provide a well established and widely accepted description scheme for assessing the quality of software products [12]. One of the major contributions provided by McCall's model is the consideration of relationships between quality characteristic and metrics [13]. There has been a criticism that not all metrics are objective and the issue of product functionality is not considered. Bohem model [14] is similar to the McCall model in that it represents a hierarchical structure of characteristics, each of which contributes to the total quality. In Bohem model the main emphasis is on the maintainability aspect of a software product. However, Bohem's model contains only a diagram and does not elaborate the methodology to measure these characteristics. Dromey [15] proposed a quality evaluation framework taking into consideration relationship between the attributes (characteristics) and sub-attributes (sub-characteristics) of the quality. This model suffers from lack of criteria for measurement of software quality and it is difficult to see how it could be used at the beginning of the lifecycle to determine the user needs. The FURPS [16-19] quality model covers *Functional requirements* and Non-functional requirements but fails to take into account the software product's maintainability, which may be important criterion for application development, especially for component based software systems (*CBSS*). The Bayesian belief network (*BBN*) is a special category of graphical model, which represents quality as root node and other quality characteristics via directed arrows [20-21]. This model is useful to manipulate and represent complex quality model that cannot be established using conventional methods. However, this model fails to evaluate fully software product due to involvement of lack of criteria. Different perspectives of software quality can be represented by star model. Even though it considers various viewpoints of quality it does not evaluate fully software product due to involvement of lack of criteria. The ISO 9126 [22] is a part of ISO

9000 standard, which is the most important standard for quality assurance. In this model, the totality of software product quality attributes is classified in a hierarchical tree structure of characteristics and sub-characteristics. This model is too generic to be applicable to component based environments.

Bertoa's model [23] is a well known initiative to define the attributes that can be described by *COTS* vendors. In the model *portability* and *fault tolerance* characteristics disappear together with the *stability* and *analyzability* sub-characteristics. Two new sub-characteristics: *compatibility* and *complexity* are added in this model. Although this model presents a good description on quality characteristics but *portability* and *fault tolerance* which have been eliminated are very significant to *COTS* components. The Alvaro's model [24] is similar to Bertoa's model but provides better footprints as the model has introduced a number of components specific quality characteristics or sub-characteristics like *self-contained*, *configurability* and *scalability*. The purpose of the model is to determine which quality characteristic should be considered for the evaluation of software component [25]. *Reusability* is important for the reason that software factories have adopted component based approaches on the premise of reuse. The *maintainability* and *analyzability* sub-characteristics have been removed from ISO 9126. A high level characteristics 'Business' have also been added with following sub-characteristics: *development time*, *cost*, *time to market*, *targeted market*, *affordability*. Alvaro's model also has some drawbacks. Firstly, *reusability* has been treated as quality attribute rather than quality factor. Secondly, the ambiguous definition of *scalability* which is related to only data volume and not the maximum number of components, the component can interact with other components without reducing performance. In Rawedah's model [26] standard set of quality characteristics suitable for evaluating *COTS* components along with newly defined sets of sub-characteristics associated with them were identified. The sub-characteristics *fault tolerance*, *configurability*, *scalability* and *reusability* have been removed. New characteristic manageability with sub-characteristics *quality management* has been added. The model also attempts to match the appropriate type of stakeholders with the corresponding quality characteristics. It can be easily notice that the sub-characteristics that have been removed are significant to components. It can be clearly seen that still a comprehensive component specific quality model is not available.

3 Proposed Software Component Quality Model (SCQM)

Based on critical literature survey [13-31] we propose component specific quality model (*SCQM*) which overcomes the shortcomings of existing quality models. At the highest level the *SCQM* consists of eight characteristics – *functionality*, *reliability*, *usability*, *efficiency*, *maintainability*, *portability*, *reusability* and *traceability*. A unified measure for attributes is taken on a level of satisfaction scale (LOS) from 1 – 5, see table 1. The detail description of the *SCQM* characteristics is as follows:

3.1 Functionality

This characteristic indicates the ability of a component to provide the required services and functions, when used under the specified conditions. It directly matches

with the functionality characteristic of ISO 9126 but with a set of different sub-characteristics. More specifically, *security* and *suitability* sub-characteristic retain their meaning with the only exception that the term *suitability* is replaced with the term *completeness* as it reflects better scope. Since the components are meant to be reuse, *interoperability* is moved under the category of *reusability* characteristics (not in ISO 9126). *Reusability* is treated as a separate. Also, *accuracy* is moved to the *reliability* characteristic (*Result set* sub-characteristic) since we consider Accuracy to be a feature of *reliability* instead of *functionality*. We have included *self-containment* which is the intrinsic property of a component. Finally, for the evaluation reason, *compliance* has been removed temporarily because currently there are still no official standards to which component must adhere to. Following sub-characteristics contribute to *functionality*:

- *Self-containment*: *Self-contained* is an intrinsic property of a component and it means component is encapsulated with well-defined interfaces and can be executed independently with minimal outside support [27]. Following attributes contribute to Self-containment of a component

- *Preconditions and postconditions*: Well defined interfaces have contacts that are described by the presence of pre-conditions and postconditions [27].

Measure = (Boolean scale: If value is 1 then LOS otherwise value is set to 0)

- *Modularity*: It indicates the degree of independence of the component, which is the degree of functional cohesion. It can be measured by the ratio of *total number of functions provided by the component itself without external support* to the *total number of functions provided by the component*.

Measure = (Ratio normalized to LOS)

- *Security*: It indicates the ability of a component to control the unauthorized access to the services provided to it. It also deals with whether a security failure of a component will result to a system wide security failure. Following attributes contribute to security of a component:

- *Control access*: It indicates the ability of a component to provide control access mechanisms like *authentication* and *authorization*.

Measure = (Boolean scale: If value is 1 then LOS otherwise value is set to 0)

- *Data encryption*: It indicates the ability of a component to provide data encryption mechanisms to secure data it handles.

Measure = (Boolean scale: If value is 1 then LOS otherwise value is set to 0)

- *Auditing*: It indicates the ability of a component to keep track of user actions and any unauthorized access.

Measure = (Boolean scale: If value is 1 then LOS otherwise value is set to 0)

- *Privilege intensification*: It indicates the ability of a component to identify any flaw in the component which may leads to privilege intensification and hence to system security breach.

Measure = (Boolean scale: If value is 1 then LOS otherwise value is set to 0)

- *Completeness*: It indicates the ability of a component to fit into user (re-user) requirements. This provides the overall idea of the level to which component

covers the needs of the users in terms of services offered. Following attributes contribute to completeness of a component:

- *User Satisfaction*: It indicates the ability of a component to meet re-user requirements by offering services as per re-user needs. Here, re-users are not the component providers but are system developers and integrators.

$$\text{Measure} = (\text{LOS})$$

- *Service Excitability*: It indicates the ability of a component to provide more than required/expected related service. It can be measured by the ratio of *total number of functions required by the user* to the *total number of functions provided by the component itself*.

$$\text{Measure} = (\text{Ratio normalized to LOS})$$

Table 1. Level of satisfaction (LOS) scale

S.No.	Description	Scale (LOS)
1	Very Low Satisfaction (VLS)	1
2	Low Satisfaction (LS)	2
3	Moderate Satisfaction (MS)	3
4	High Satisfaction (HS)	4
5	Very High Satisfaction (VHS)	5

3.2 Reliability

Reliability is the capability of a component to maintain a specified level of performance when used in stated conditions in a stated period of time. It also indicates the ability of a component to return correct results in a quality manner. This characteristic is directly related to ISO 9126 but with minor modifications in the sub-characteristics notions. *Maturity* is renamed as *service maturity* in order to encompass the meaning of contributed attributes such as – *error prone*, *error handling*, *recoverability*, and *availability*. It is to be noted that the sub-characteristics *recoverability* and *fault tolerance* mentioned in ISO 9126 is retained but included in *service maturity* as measurable attributes. Also, a new sub-characteristic *Outcome Set* is introduced which asserts the correctness and quality of the results returned by the component. Following sub-characteristics contribute to Reliability of a component:

- *Service Maturity*: It expresses the level of confidence that the component is free from errors. It also indicates the time when the component is available with the services and the ability to recover from failure. Following attributes contribute to service maturity of a component
 - *Error prone*: It indicates the ability of a component to identify how much it is prone to system errors (complete system failure) and the frequency and the relative importance of those errors. It can be measured by the number of errors occurring per unit of time. Less number of the errors means less errors prone to system failure.

$$\text{Measure} = (\text{Ratio normalized to LOS})$$

- *Error Handling*: It indicates the ability of a component to provide handling mechanisms to handle if any error encountered.

Measure = (Boolean scale: If value is 1 then LOS otherwise value is set to 0).

- *Recoverability*: It indicates the ability of a component to recover when error occurs (fault tolerance). It also indicates that after recovery whether any data or system loss happens.

Measure = (Boolean scale: If value is 1 then LOS otherwise value is set to 0).

Availability: It indicates the ability of a component to be available for offering services. It is measure on the basis of duration of its availability i.e. the average uptime of the component without serious errors or crashes.

Measure = (LOS)

- *Outcome Set*: It indicates the ability of a component to provide correct and quality results. In addition, it indicates that the component support transaction based computation. Following attributes contribute to Outcome set of a component:
 - *Correctness*: It indicates the ability of a component to return correct results. In addition, it also indicates the quality of results in terms of computational precision and accuracy. It is measure as a ratio of '*as expected*' results to the *total number of results*.

Measure = (Ratio normalized to LOS)

- *Transactional*: It indicates the ability of a component to provide transactional processing i.e. rollback facility if transaction fails.

Measure = (Boolean scale: If value is 1 then LOS otherwise value is set to 0).

It is to be noted that reliability can also be assessed by measuring the frequency and severity of failures, the accuracy of output results, the mean time between failures and the ability to recover from failure and the predictability of the program [26].

3.3 Usability

The usability characteristic represents the significant difference in meaning between ISO 9126 and component quality model. The difference lies in the fact that in component specific quality model the users are considered primarily system developers, who handle the integration of the components to their systems. Although most of the sub-characteristics remain the same in term of naming (only *understandability* is renamed to *help* mechanisms). Help mechanisms need to target different people. Some look for "how to" administer component and some are interested in understanding "*interfaces*" support. From the sub-characteristics list of *usability*, *attractiveness* and *compliance* have been removed as end-users are the system developers and not the one which interact with complete system and due to lack of standards compliance is not possible. Additionally, approachability as sub-characteristic of *usability* is introduced in order to represent the need for effective identification and retrieval of the desired component [28]. The description of *usability* sub-characteristics is as follows:

- Help Mechanisms

This sub-characteristic denotes availability and effectiveness of help facility for the usage of the component. Following attributes are used to compute the level of *help mechanisms* supported by the component:

- *Help System*: It indicates availability and completeness of the help files for the system.

Measure = (Boolean scale: If value is 1 then LOS otherwise value is set to 0).

- *Manuals*: It provides the presence of *user manual*, *administration manual* and *installation manual*

Measure = (Boolean scale: If value is 1 then LOS otherwise value is set to 0).

- *Tutorials and Demos*: It indicates presence of *tutorials* and *demos* to support the usage of component.

Measure = (Boolean scale: If value is 1 then LOS otherwise value is set to 0).

- *Support Tools and services*: It denotes the presence of other supporting tools and services to enhance the usage of component such as online help using chat services or telephonic help, expert training modules or sessions etc.,

Measure = (Boolean scale: If value is 1 then LOS otherwise value is set to 0).

- Learnability

The sub-characteristic indicates the time needed by a user to learn *how to use*, *configure* and *administer* a component. Here, the users are considered to have an average experience in component related projects. The contributing attributes for *learnability* are as follows:

- *Time to use*: It indicates the time needed by an average user to learn how a component works and how it can be used in a software system.

Measure = (Boolean scale: If value is 1 then LOS otherwise value is set to 0).

- *Time to configure*: It denotes the time required for an average user to learn how to configure the component for its usage.

Measure = (Boolean scale: If value is 1 then LOS otherwise value is set to 0).

- *Time to administer*: It indicates the time required for an average user to administer the component.

Measure = (Boolean scale: If value is 1 then LOS otherwise value is set to 0).

- Operability

It indicates the level of effort required to *operate*, *administer* and *customize* the component. Following attributes contribute to *operability*:

- *Operation effort*: It indicates the effort required to operate the component. It is heavily dependent upon the suitability of the component on the assigned tasks. For example, whether a component requires some manual tasks to be performed for its operation or not.

Measure = (LOS).

- *Administration effort*: It denotes the effort required to administer a component. The functioning is similar to *operation effort* but mainly focus on administration related tasks.

Measure = (LOS).

- *Customizability effort*: It indicates the effort required to customize a component for its per-defined interfaces.

Measure = (LOS).

- **Approachability**

It indicates the capability of a component to be searched by its users for its usage through search mechanisms. The contributing attributes for *approachability* are as follows:

- *Directory listings*: It indicates the simplicity of finding a component. Dedicated sites on the World Wide Web or special software magazines are the source of components listings and can be used for such findings. It can be measured as the *Ratio of popular directory listings that the component is marketed to the total number of popular directory listings.*

Measure = (Ratio normalized to LOS)

- *Search & Fetch*: It denotes the simplicity in searching and fetching a component. For example, whether any pre-requisites are required for approving download of component (trial version) such as registration, software or hardware, training etc., or not.

Measure = (LOS)

- *Classification*: It denotes the supportability of the classification scheme by a component. For example, the component can be classified according to packaging or platform specific views etc.

Measure = (LOS)

- *Marketing information*: It indicates the ability to understand the component capabilities without actually installing it. It deals with the information the vendor has provided as part of marketing strategies.

Measure = (LOS)

3.4 Efficiency

It is the capability of a component to provide appropriate performance, relative to the amount of resources used under stated conditions. It corresponds to the efficiency characteristics of ISO 9126. Following sub-characteristics contributes to efficiency of a component:

- *Time Behavior*: It indicates the time difference between component's method invocation and getting its response. Following attributes contribute to time behavior of a component:

- *Throughput*: It indicates the ability of a component how fast it serves requests and provides results over a given period of time. It is measured as the ratio of *number of successful served requests per unit time*.

Measure = (Ratio normalized to LOS)

- *Capacity*: It indicates the ability of a component to serve to number of users simultaneously without degrading performance level. It is measured by the number of users supported by the component. More the users better the capacity.

Measure = (LOS)

- *Concurrency*: It indicates the ability of a component to provide synchronous or asynchronous invocation.

Measure = (Boolean scale: If value is 1 then LOS otherwise value is set to 0)

- *Resource Behavior*: It indicates the ability of a component to utilize resource to run itself. attributes contribute to resource behavior of a component:
 - *Memory utilization*: It indicates the amount of memory needed by a component to operate. It is measured by the number of kilobytes of RAM required for its operation. More kilobytes means component utilizes high memory.

Measure = (LOS)

- *Processor utilization*: It indicates the amount of processing time (CPU cycle) needed by a component to operate. It is measured by the average number of CPU cycles required for its operation. More CPU cycles means component utilizes high processing.

Measure = (LOS)

- *Disk utilization*: It indicates the amount of disk space needed by a component to operate. It includes both the disk space for installing the component and other supported materials (such as documentation, help files etc.), as well as the disk space used temporarily during execution time. It is measured by the number of kilobytes of disk required for its operation. More kilobytes means component utilizes high disk space.

Measure = (LOS)

3.5 Maintainability

Maintainability is the effort required to replace a COTS component with the corrected version and to migrate an existing, software component from a current component based software system to a new version of the system [29]. The description of *maintainability* sub-characteristics is as follows:

- Customizability

It evaluates the capability of a component to be customized according to the user needs. Following attributes contribute to customizability of a component:

- *Parameterization*: It indicates the number of parameters offered for change by the component with the number of provided interface. Black-box parameterization is measured by *the ratio of number of parameters available for change to the total number of interfaces supported (offered) by a component*.

Measure = (LOS)

- *Adaptability*: It indicates the ability of a component to adapt itself to a changing environment at runtime.

Measure = (LOS)

- *Change control capability*: It indicates the ability of a component to make user aware of current version of a component.

Measure = (LOS)

- *Priority*: It indicates the capability of a component to provide prioritize service, i.e. some of its function assume priority at runtime (ISO/IEC, 1991).

Measure = (Boolean scale: If value is 1 then LOS otherwise value is set to 0)

• Testability

It examines the features of a component that can be tested by supporting test cases, tools or test suites. Following attributes contribute to testability of a component:

- *Start up self test*: It indicates the capability of a component to test itself and environment for operation.

Measure = (Boolean scale: If value is 1 then LOS otherwise value is set to 0)

- *Trial version*: It denotes the ability of a component to support trial version in order to facilitate user to perform test for the functionality support by a component. Associated with trial version is its timeline usage scale – limited or full version, whether a component is a freeware (f), shareware (s), or commercial (c), see table 2.

Measure = (LOS).

Table 2. Component trial version type and scale

S.No.	Trial version Limited/full	Type {f, s, c}	Scale (1-5)
1	Limited	F	{VL, L, M, H, VH}
2	Limited	S	{VL, L, M, H, VH}
3	Limited	C	{VL, L, M, H, VH}
4	Full	F	{VL, L, M, H, VH}
5	Full	S	{VL, L, M, H, VH}
6	Full	C	{VL, L, M, H, VH}

- *Test suite provided*: It indicates the presence or absence of a test suite on a
- Measure = (Boolean scale: If value is 1 then LOS otherwise value is set to 0)**

- *Test materials*: It denotes the existence of other useful test materials like demos, gray code (some level of visible code), logical and data flow diagrams and test cases.

Measure = (Boolean scale: If value is 1 then LOS otherwise value is set to 0)

• Changeability

It indicates the effort needed to modify a component at ease as per requirements. Following attributes contribute to changeability of a component:

Upgradeability: It denotes the ability of a component to be upgraded to a new version at ease. If upgradation requires manual support or intervention leading to system downtime then this will result into negative impact.

Measure = (LOS).

Debugging: It denotes the ability of a component to support debugging (functional). It evaluates the efficiency of an error messages returned by the component in order to understand the erroneous functionality of component and then corrects it. *The ratio of descriptive and understandable errors to the total number of errors provided by a component can be used as a measure for debugging capability of a component.*

Measure = (Ratio normalized to LOS)

Backward compatibility: It indicates whether a new component is compatible (backward) with previous versions or not. If a component is not backward compatible then re-user has to re-write the software system to achieve full functionality in order to accommodate changes.

Measure = (LOS)

3.6 Portability

This characteristic indicates the ability of a component to be transferred from one environment to another with little or no modification. It corresponds to ISO9126 model in the meaning but two of its sub-characteristics have been transferred to maintainability characteristics. *Replaceability* is renamed as *backward compatibility*, has been transferred to sub-characteristic *changeability* of *maintainability*. Similarly, *adaptability* has been transferred to sub-characteristic *customizability* of *maintainability*. *installability* characteristic is retained. In addition, *deployability* sub-characteristic is added as a component instance may be deployed only once but installed any number of times [28]. Following sub-characteristics contribute to portability of a component:

- *Installability*: It is the capability of a component to be installed on different platforms. Following attributes contribute to installability of a component:

- Documentation: It indicates the ability of a component to provide supportive installable documentation or not.

Measure = (Boolean scale: If value is 1 then LOS otherwise value is set to 0)

- Complexity: It is the ability of a component which shows how complex it is to carry out the installation of a component.

Measure = (LOS).

- *Deployability*: It is the capability of a component to be deployed on different platforms. Following attributes contribute to deployability of a component:

- Documentation: It indicates the ability of a component to provide supportive delployable documentation or not.

Measure = (Boolean scale: If value is 1 then LOS otherwise value is set to 0)

- Complexity: It is the ability of a component which shows how complex it is to carry out the deployment of a component.

Measure = (LOS).

3.7 Reusability

It is a critical characteristic in the development of component based software system [30], as component can be reused to create more complex applications. Since Reusability is a driving force for component markets we have included this as a separate characteristic. Reusability improves productivity, maintainability and overall quality of a system. Following sub-characteristics contribute to Reusability of a component:

- *Interoperability*: It is the ability of a component to be reused. The evaluation of interoperability is of prime importance since reusability aspect is the cornerstone of CBSS success. Following attributes contribute to Interoperability of a component:

- *Platform Independence*: It indicates the ability of a component to be used in different component models such as CORBA, COM/DCOM and other similar models. It is measured as a ratio of *number of platforms/models supported* to the *number of most used platforms/models*.

Measure = (Ratio normalized to LOS)

- *Operating System Independence*: It indicates the ability of a component to be used in different operating systems such as Windows, LINUX, UNIX, AIX etc., and other similar operating systems. It is measured as a ratio of *number of operating systems supported* to the *number of most used operating systems*.

Measure = (Ratio normalized to LOS)

- *Hardware Compatibility*: It indicates the ability of a component to be used in different types of hardware such as personal computers, laptops, PDAs and other similar hardware. It is measured as a ratio of *number of hardware supported* to the *number of most used hardware*.

Measure = (Ratio normalized to LOS)

- *Data Open-format Compatibility*: It is the ability of a component to output data that is compatible with the well-known formats.

Measure = (Boolean scale: If value is 1 then LOS otherwise value is set to 0)

- *Generality*: It is the capability of a component to be generic so that it can be reused in number of applications. Following sub-characteristics contribute to generality of a component
 - *Domain abstraction*: It is the ability of a component to be reused across several domains related to specific functionality that the component offers.

Measure = (Boolean scale: If value is 1 then LOS otherwise value is set to 0)

- *Reuse maturity*: It is the ability of a component to show/support the reusability record and documentation for future reference.

Measure = (Boolean scale: If value is 1 then LOS otherwise value is set to 0)

3.8 Traceability

This characteristic expresses the extent of a component's built in capacity of tracking the status of component attributes and component behavior. According to [31] traceability refers to the ability to show, at any time, where an item is, its status, and where it has been. It can be noticed that while reconfiguring a component for changed/improved functionality, maintainers must perform a full cycle of product evaluation, integration and testing. Replacing an older version of a component with newer version can also create problems such as security violation, resource usage and performance usage degradation etc. This characteristic is not present in ISO 9126 but we felt that *traceability* of the component is important so as to validate any violation during modification/replacement of a component. So we treated *traceability* as a separate characteristic. Following sub-characteristics contribute to traceability of a component:

- *Behavior traceability* - It is the ability of a component (black box) to track its external behaviors. The major purpose is to track component public visible data or object states, visible events, external accessible functions and the interactions with other components. Following attributes contribute to behavior traceability
 - *Performance trace* - It is the ability of a component to records the performance data and benchmarks for each function of a component in a given platform and environment.

Measure = (LOS)

- *State trace* - It tracks the object states or data states in a component.

Measure = (LOS)

- *Controllability Trace* - It is the ability of a component that refers to the extent of the control capability to facilitate the customization of its tracking functions. Following attributes contribute to controllability trace
 - *Customization trace*: It is the ability of a component to control and set up various tracking functions such as turn-on and turn-off of any tracking functions and selections of trace formats and trace repositories.

Measure = (LOS)

- *Error trace* - It records the error messages generated by a component. The error trace supports all error messages, exceptions, and related processing information generated by a component.

Measure = (LOS)

The proposed component quality model does not discuss *business* characteristics (specific to market) as it does not certify to be quality characteristic specific to component (in quality context).

4 Conclusion

The proposed Quality model (*SCQM*) is a first step towards research effort in the direction of building a software component quality assurance framework. The model was designed in such a way as to overcome the drawbacks of existing models. Research is undergoing to develop capability maturity model for software components that can be mapped to the proposed component model. Our future work will try to incorporate emerging needs of (re)users as the software technology advances such as mobile computing and pervasive computing. We are also focusing on developing dedicated web application providing XML schema, evaluation tool and structured feedback mechanisms in order to achieve world-wide evaluation. This may results for evaluators to discuss and share their experiences regarding usage and difficulty level in using *SCQM*.

References

- [1] Tran, V., Liu, B.D., Hummel, B.: Component-based Systems Development: Challenges and Lessons Learned, Software Technology and Engineering Practice. In: Proceedings of the Eighth IEEE International Workshop on ICASE, pp. 452–462 (1997)
- [2] Raje, R., Bryant, B., Auguston, M., Olson, A., Burt, C.: A Unified Approach for the Integration of Distributed Heterogeneous Software Components. In: Proceedings of the 2001 Monterey Workshop Engineering Automation for Software Intensive System Integration, Monterey, California, pp. 109–119 (2001)
- [3] Kallio, P., Niemela, E.: Documented quality of COTS and OCM components. In: Proceedings Fourth ICSE Workshop on Component-Based Software Engineering, Toronto, Canada, pp. 111–114 (2001)
- [4] Preiss, O., Wegmann, A., Wong, J.: On Quality Attribute Based Software Engineering. In: Proceedings of the 27th Euromicro Conference, pp. 114–120 (2001)
- [5] Szyperski, C.: Component Object-Oriented Programming. Addison-Wesley, Reading (1998)
- [6] Andreou, S.A., Tziakouris, M.: A Quality Framework for Developing and Evaluating Original Software Components. Information and Technology 49, 122–141
- [7] Upadhyay, N., Deshpande, B.M., Agrawal, V.P.: MACBSS: Modeling and Analysis of Component Based Software System. In: Proceedings of IEEE World Congress on Computer Science and Information Engineering, Los Angeles, USA, pp. 595–603 (2009)
- [8] Rakic, M., Medvidovic, N.: Increasing the Confidence in Off-The-Shelf Components: a Software Connector-Based Approach. In: Proceedings of the Symposium on Software Reusability, pp. 11–18 (2001)
- [9] Mann, S., Borusan, H., Grobe-Rohde, M., Mackenthun, R., Sunbul, A., Weber, H.: Towards a Component Concept for Continuous Software Engineering, Fraunhofer ISST, Technical Report (2000)
- [10] Gao, W.: Testing and Quality Assurance of Software Components. Arctech Publishing House, Boston (2003)

- [11] Parminder, K., Hardeep, S.: Certification Process of Software Components. ACM SIG-SOFT Software Engineering Notes 33 (2008)
- [12] Behkamal, B., Kahani, M., Akbari, K.M.: Customizing ISO 9126 Quality Model for Evaluation of B2B Applications. *Information and Software Technology* 51, 599–609
- [13] Fizpatrick, R.: Software Quality definitions and strategic issues, Technical Paper, Staffordshire University (1996)
- [14] Bohem, B.W., Brown, J.R., Kaspar, H., Lipow, M., McLeod, G., Meritt, M.J.: *Characteristics of Software Quality*. North Holland Publishing, Amsterdam (1978)
- [15] Dromey, R.G.: A Model for Software Product Quality. *IEEE Transactions on Software Engineering* 21, 146–162 (1995)
- [16] Khosravi, K., Guehneuc, Y.: A Quality Model for Design Patterns, Technical report 1249, University of Montreal (2004)
- [17] Jacobson, I., Booch, G., Rumbagch, J.: *The Unified Software Development Process*. Addison-Wesley, Reading (2000)
- [18] Krutchen, P.: *The Rational Unified Process: An Introduction*. Addison-Wesley, Reading (2000)
- [19] Grady, R.: *Practical Software Metrics for Project Management and Process Improvement*. Prentice-Hall, Englewood Cliffs (1992)
- [20] Stefani, A., Xenos, M., Stavrinoudis, D.: Modeling E-Commerce Systems' Quality with Belief Networks. In: *Proceedings of International Symposium on Virtual Environments, Human- Computer Interfaces, and Measurement systems*, Switzerland (2003)
- [21] Stefani, A., Stavrinoudis, D., Xenos, M.: Experimental Based Tool Calibration used for Assessing the Quality of E-commerce Systems. In: *Proceedings of the First IEEE International Conference on E-Business and Telecommunication Networks*, Portugal, pp. 26–32 (2004)
- [22] ISO International Organization for Standardization.: ISO 9126-1:2001 Software engineering-Product quality, Part 1: Quality Model (2001)
- [23] Bertoa, M., Vallecillo, A.: Quality Attributes for COTS Components. In: *Proceedings of the 6th International ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering (QAOOSE)*, Spain (2002)
- [24] Alvaro, A., Almeida, A.S.: Towards A Software Component Quality Model. In: *Proceedings of the 5th International Conference on Quality Software* (2005)
- [25] Alvaro, A., Almeida, A.S., Meira, S.R.: Quality attributes for a Component Quality Model. In: *Proceedings of the 10th International Workshop on Component Oriented Programming (WCOP) in conjunction with the 19th European Conference on Object Oriented Programming (ECOOP)*, Glasgow, Scotland (2005)
- [26] Raweshdah, A., Matakah, B.: A New Software Quality Model for Evaluating COTS Components. *J. of Comp. Sc.* 2, 373–381 (2006)
- [27] Szyperski, C., Dominic, G., Stephen, M.: *Component Oriented Programming-Beyond Object Oriented Software*, 2nd edn. Addison Wesley and ACM Press, New York (2002)
- [28] Hansen, W.J.: An Original Process and Terminology for Evaluating COTS Software, <http://www.sei.cmu.edu/staff/wjh/Qesta.html> (accessed March 25, 2010)
- [29] Gao, J., Gupta, K., Gupta, S., Shim, S.: On Building Testable Software Components. In: Palazzi, B., Gravel, A. (eds.) *ICCBSS 2002*. LNCS, vol. 2255, pp. 108–121. Springer, Heidelberg (2002)
- [30] Jon, H.: Component Primer. *Communication of ACM* 43, 27–30 (2000)
- [31] Schmauch, C.H.: *ISO 9000 for Software Development: Revised Edition*, American Society for Quality (1995)

Colour Image Encryption Scheme Based on Permutation and Substitution Techniques

Narendra K. Pareek¹, Vinod Patidar², and Krishan K. Sud²

¹ University Computer Centre, Vigyan Bhawan,
M L Sukhadia University, Udaipur-313 002, India
npareek@yahoo.com

² Department of Physics, School of Engineering
Sir Padampat Singhania University, Bhatewar, Udaipur- 313 601, India
vinod.patidar@spsu.ac.in, kksud@yahoo.com

Abstract. In this paper, a new lossless image encryption scheme using a secret key of 128-bit size is proposed. In the algorithm, image is partitioned into several key based dynamic blocks and each block is passed through the eight rounds of permutation as well as substitution process. In permutation process, sequences of block pixels are rearranged within the block by a zigzag approach whereas block pixels are replaced with another by using difference transformation of row and column in substitution process. Due to high order of confusion and diffusion, attacks like linear and differential cryptanalysis are infeasible. The experimental results show that the proposed technique is efficient and has high security features.

Keywords: Cipher, Image encryption, Permutation, Secret key, Substitution.

1 Introduction

Security of image data has become increasingly important for many applications like video conferencing, secure facsimile, medical, military applications etc. It is hard to prevent unauthorized people from eavesdropping in any communication system including internet. To protect information from unauthorized people, mainly two different technologies- digital watermarking and cryptography, is being used. These two technologies could be used complementary to each other. In secured communications using cryptography, which is the main focus of the present work, the information under consideration is converted from the intelligible form to an unintelligible form at transmitter end. Encryption process scrambles the content of data such as text, image, audio, and video to make the data unreadable, invisible or incomprehensible during transmission. The encrypted form of the information is then transmitted through the insecure channel to the destination. At the intended recipient side, the information is again converted back to an understandable form using decryption operation. When same key is used to guide both encryption and decryption operations, such algorithms are grouped under private key cryptography. Numerous private key image ciphers

based on different techniques are available in literature. In the following paragraph, we discuss in brief some of the recent work on image cryptographic schemes.

An image encryption algorithm based on high dimensional chaotic systems was suggested by Li and Han [1]. In this encryption algorithm, authors used generalized cat map to diffuse an image data whereas Lorenz system was employed to confuse the diffused image data. Further, Çokal and Solak [2] pointed out security weakness of chaos-based image encryption algorithm proposed by Guan et al. [3]. They disclosed all the secret parameters of the algorithm by applying chosen-plaintext as well as known-plaintext attacks. Later, an image encryption algorithm based on composite discrete chaotic system was developed by Cheng et al. [4] in which sort transformation was used in confusion process to modify image data. Sun et al. [5] designed a chaotic image encryption algorithm in which both control parameter as well as initial value of chaotic sequence was used as secret keys. Further, Chen and Zhang [6] presented a chaotic image encryption scheme based on Henon and Arnold cat map. In this algorithm, Arnold cat map was used to shuffle the positions of the image pixels whereas Henon's map was used to encrypt the resultant shuffled image pixel by pixel. Rhouma et al. [7] proposed a colour image encryption scheme based on one-way coupled-map lattices (OCML) in which a 192-bit long external secret key was used to generate both initial conditions and the control parameters of the OCML. Yang et al. [8] suggested a chaotic block cipher taking plaintext block of 128-bits size. In the algorithm, secret key was formed by combination of 128-bit key, initial condition and control parameter of logistic map. The algorithm includes the process of permutation, eight rounds and transformation. Further, Xiao et al. [9] analysed the chaos-based image encryption algorithm suggested by Guan et al. [3] and pointed out its flaws. They also suggested a few points to improve the same algorithm. A block encryption scheme based on dynamic substitution boxes (S-boxes) was proposed by Wang et al. [10]. Authors discussed a method for generating S-boxes based on iterating the tent map. Each plaintext blocks were encrypted using S-boxes, 32 rounds of substitution and left cyclic shift. Further, Patidar et al. [11] proposed a lossless symmetric image cipher based on chaotic standard and logistic maps. In the algorithm, the initial condition, system parameter of the chaotic standard map and number of iterations together constituted the secret key of image cipher. To encrypt an image, it passed through four different rounds where contents of image were permuted and substituted. Rhouma et al. [12] cryptanalysed the algorithm [11] successfully using the known-plaintext and known-ciphertext attack. Further, an image encryption scheme based on 3D Baker map with dynamical chaotic sequence was designed by Tong and Cui [13] which was later studied by Li et al. [14]. Li et al. [14] pointed out security weakness of algorithm [13] which are - insecure against a differential chosen-plaintext attack, a number of weak keys and some equivalent keys, insensitive to changes in plain-images. In a chaotic image encryption algorithm proposed by Wang et al. [15], dynamic control parameters were used rather than fixed value of it. In the algorithm, control parameters were used in the permutation process and the keystreams employed in the diffusion process were generated from two chaotic maps. Next, Wong et al. [16] proposed a new diffusion mechanism for image encryption using table lookup and swapping techniques. Arroyo et al. [17] studied the chaotic image encryption scheme proposed by Gao and Chen [18] and could cryptanalysed the same with a chosen-plaintext attack. Further, an image encryption scheme based on coupled map

lattices with time delay was suggested by Tang et al. [19]. They used discretized tent map in algorithm to shuffle the positions of image pixels and delayed coupled map lattices to confuse the relationship between the plain image and the cipher image. Wang et al. [20] developed a block image encryption scheme using combination of confusion and diffusion. In this encryption algorithm, Baker map was used to generate a pseudo-random sequence and several one dimensional chaotic maps were dynamically selected to encrypt blocks of image. Alvarez and Li [21] pointed out security weakness of an image encryption scheme based on a logistic-like new chaotic map proposed by Gao et al. [22] and cryptanalysed the same successfully through differential attack. Next, Patidar et al. [23] improved their earlier developed chaotic image cipher [11] to make it robust against the cryptanalytic attacks. A chaos-based image encryption algorithm based on coupled nonlinear chaotic map was proposed by Mazloom and Moghadam [24]. In the algorithm, 240-bit long secret key was used to generate the initial conditions and control parameters of the chaotic map by making some algebraic transformations to the key. Further, an image encryption algorithm with combined permutation and diffusion was proposed by Wang et al. [25]. In the algorithm, authors partitioned an image into blocks of pixels and then, spatiotemporal chaos was employed to shuffle the blocks and to change the pixel values. Next, Xiao and Xia [26] proposed an image encryption scheme in which sort transformation based on nonlinear chaotic algorithm was used to shuffle the positions of image pixels and hyper-chaos was used to change the grey values of the shuffled image.

In the literature, most of the existing image encryption schemes are based on chaotic system. In this paper, a non-chaos based image encryption scheme using secret key of 128-bit size is presented. In the algorithm, image is partitioned into several dynamic blocks and each block is passed through the eight rounds of permutation process. In each round, block size is kept different which depends on the secret key used in the algorithm. In permutation process, sequences of block pixels are rearranged within the block by a zigzag approach as shown in the Fig 2. Further, blocks are resized and pass through the eight rounds of substitution process. In each round of substitution process, size of each block is key dependent and may be different from block size used in permutation process. In substitution process, block pixels are replaced with another by using difference transformation of rows and column pixels. The rest of the paper is organized as follows. In Section 2, details of different components used in the proposed algorithm as well as complete encryption algorithm are introduced. Security analysis and simulation results of the proposed image encryption scheme are presented in Section 3. Finally, Section 4 concludes the paper.

2 Proposed Encryption/Decryption Algorithm

Image data have strong correlations among adjacent pixels forming intelligible information. To encrypt the image, this intelligible information needs to be reduced by decreasing the correlation among the pixels. The proposed encryption algorithm does this by scrambling the pixels of the image as well as modifying the pixel values of the resultant image. In the following paragraph, we discuss in detail the functions of different components used in the proposed algorithm.

2.1 Plain Image Block Size

In each round of permutation and substitution process, plain image pixels are divided into several non-overlapping squared dynamic blocks and their size (B_r) depends on the secret key used in the algorithm. Both (1) and (2) are used to decide the plain image block size in permutation and substitution process respectively.

$$B_r = \sum_{p=1}^4 K_{(4*(r-1)+p)} \quad (\text{Permutation process}) \tag{1}$$

$$B_r = \sum_{p=1}^4 K_{(4*(8-r)+p)} \quad (\text{Substitution process}) \tag{2}$$

where K_i and B_r are the i^{th} subkey and block size in r^{th} round respectively.

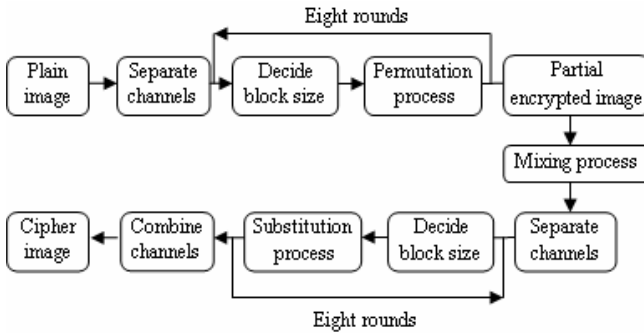


Fig. 1. Block diagram of the proposed image cipher

2.2 Permutation Process

In the permutation process, pixels of each dynamic block are rearranged within itself by a zigzag path. For example, a path shown in Fig 2 is used to rearrange the pixels of a block of 8x8 size. In Fig 2, if we start traversing the path from the pixel location (7,6), then stop traversing at the pixel location which is previous to the start one i.e. (6,7). During traversing process, the pixels, encounter in the path, are arranged sequentially row by row and column by column in the same block. Each channel (red, green and blue) of pixels passes through eight rounds of permutation. In each round, pixels are divided into several non-overlapping squared dynamic blocks as discussed in Sub section 2.1. Location of pixel (X_r, Y_r) to start traversing the path in blocks of r^{th} round is made completely secret key dependent as shown below:

$$X_r = \sum_{p=1}^3 K_{(4*(r-1)+p)} , \quad Y_r = \sum_{p=2}^4 K_{(4*(r-1)+p)} \tag{3}$$

where K_i is the i^{th} subkey.

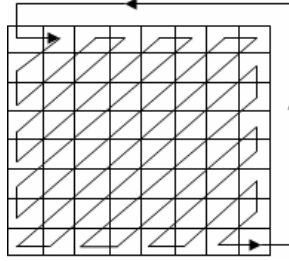


Fig. 2. A zigzag approach to scramble pixels of a block

2.3 Mixing Process

To reduce correlation between adjacent pixels in image, three different layers of pixels corresponding to red, green and blue channel of plain image are separated out and the properties of each pixel are modified using exclusive-OR operation with its previous pixel in the following way:

$$\begin{aligned}
 & i=1 \\
 & \text{for } x = 1 \text{ to } H \\
 & \quad \text{for } y = 1 \text{ to } W \\
 & \quad \quad R_{x,y} = R_{x,y} \oplus G_{x,y-1} \oplus B_{x,y-1} \\
 & \quad \quad G_{x,y} = G_{x,y} \oplus R_{x,y-1} \oplus B_{x,y-1} \\
 & \quad \quad B_{x,y} = B_{x,y} \oplus R_{x,y-1} \oplus G_{x,y-1} \\
 & \quad \text{endfor} \\
 & \text{endfor}
 \end{aligned} \tag{4}$$

where $R_{x,0}/G_{x,0}/B_{x,0} = \begin{cases} 0 & x=1 \\ R_{x-1,w}/G_{x-1,w}/B_{x-1,w} & x>1 \end{cases}$. H and W represent height and width of plain image respectively.

2.4 Substitution Process

In the substitution process, a simple computation is performed on pixels to change their properties. Each channel (red, green and blue) of pixels passes through the eight rounds. In each round, pixels are divided into several non-overlapping dynamic squared blocks as discussed in Sub section 2.1. After deciding the block size, each block is passed through two different kinds of transformations - row transformation and column transformation. For example, a row pixel of a 15x15 size block is shown in Table 1. In a row transformation, we first find largest pixel value (PV) among a row pixel which is 214 in our example and then, subtract all those pixel values of a row from PV whose pixel value is smaller to PV.

When transformation of all the rows is over, a similar operation is performed on pixels of each column obtained after row transformation. In Table 2, we have shown the pixel values for an 8x8 block before and after the complete substitution process (i.e. row transformation + column transformation).

Table 1. Procedure of a row transformation

Row pixels	86	196	134	56	23	71	162	206	213	110	8	214	45	175	32
Resultant row pixels	128	18	80	158	191	143	52	8	1	104	206	214	169	39	182

Table 2. Pixel values before and after row and column transformations

56	45	57	60	48	→	13	50	11	2	43
48	65	54	44	61		17	65	3	41	51
56	51	48	62	49		11	54	14	62	42
46	49	51	55	54		8	59	10	7	54
49	51	49	47	55		11	61	8	54	55

2.5 The Complete Encryption Algorithm

The block diagram of the proposed algorithm is illustrated in Fig 1. This system includes two major units - pixels permutation and pixels substitution unit. In the proposed image cipher, permutation and substitution processes are completely secret key dependent and a feedback mechanism is also applied in the mixing process to avoid the differential attack. We discuss the steps of algorithm in following paragraph:

1) The proposed encryption scheme uses a secret key of 128-bits size. The secret key is divided into blocks of 4-bits each referred as subkeys.

$$K=K1K2K3 \dots K32, \tag{5}$$

here, K_i 's are hexadecimal digits (0-9 and A-F).

2) Colour image is separated out into channels and each channel passes through following steps.

For round = 1 to 8 do following

- a. Decide block size (B) for current round as discussed in Sub section 2.1.
- b. Divided each channel into non-overlapping squared blocks (B).
- c. Each block passes through the permutation process as discussed in Sub section 2.2.

Endfor

3) Channels are combined to form partial encrypted image and resulting image passes through the mixing process. Further, resulting image is separated out into channels and each channel passes through following step.

For round = 1 to 8 do following

- a. Decide block size (B) for current round as discussed in Sub section 2.1.
- b. Divided each channel into non-overlapping squared blocks (B).
- c. Each block passes through the substitution process as discussed in Sub section 2.4.

Endfor

4) Resulting image is written in a file.

3 Performance and Security Analysis

An ideal image cipher should resist against all kinds of known attacks such as crypt-analytic, statistical and brute-force attacks. In this section, we discuss the security analysis of the proposed image encryption scheme such as statistical analysis, key space analysis, sensitivity analysis etc to prove that the proposed image cipher is effective and secure against the most common attacks.

3.1 Statistical Analysis

In the literature, we found that most of the existing ciphers have been successfully cryptanalyzed with the help of statistical analysis. To prove the robustness of the proposed encryption scheme, statistical analysis has been performed which demonstrates its superior confusion and diffusion properties results in a strongly resisting nature against the statistical attacks. This is done by testing the distribution of pixels of the ciphered images, study of correlation among the adjacent pixels in the ciphered image, information entropy and the correlation between the original and ciphered images.

3.1.1 Distribution of Pixels

We have analyzed the histograms of about fifty encrypted images and their corresponding plain images having widely different contents and sizes. One example of histogram analysis for well known image ‘Lena’ is shown in Fig 3. Histograms of red, blue and green channels of image (Fig 3(a)) are shown in Frames (b), (c) and (d) respectively. In Frames (f), (g) and (h) respectively, the histograms of red, blue and green channels of the encrypted image (Fig 3(e)) are shown. Comparing the histograms, we find that encryption process returns noisy images. Histograms of encrypted images are very close to uniform distribution, significantly different from that of the original image and contain no statistical resemblance to the original image. This is consistent with the perfect security defined by Shannon [27] and the encryption scheme resists against the known-plaintext attack.

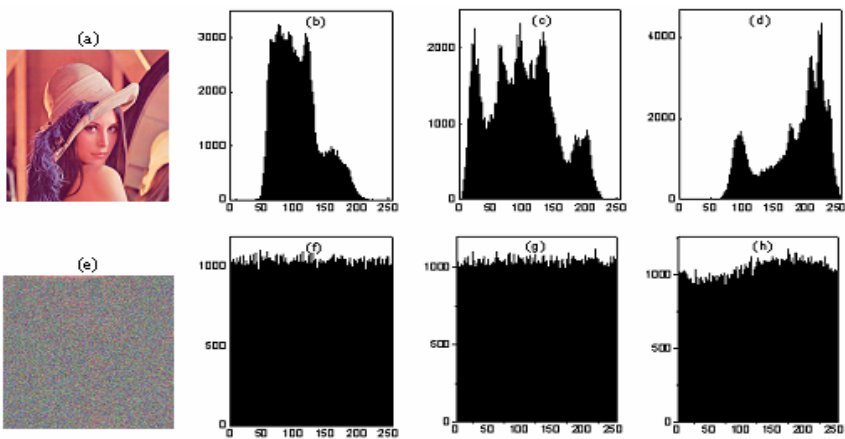


Fig. 3. Histograms corresponding to RGB channels of plain image ‘Lena’ and its corresponding encrypted image are shown

3.1.2 Correlation between Plain and Cipher Images

We have also done extensive study of the correlation between pairs of plain image and their corresponding encrypted image produced using the proposed encryption scheme by computing correlation coefficient between various colour channels of the original and corresponding encrypted images. Results for a few images are shown in Table 3. Since the correlation coefficients shown in the Table 3 are very small ($C \approx 0$), it indicates that the plain images and their corresponding ciphered images are completely independent of each other.

Table 3. Correlation coefficient between plain images and their corresponding cipher images having widely different contents

Image Size	C_{RR}	C_{RB}	C_{RG}	C_{GR}	C_{GG}	C_{GB}	C_{BR}	C_{BG}	C_{BB}
200 x 137	0.0055	-0.0130	0.0111	0.0072	-0.0212	0.0092	0.0069	-0.0235	0.0021
200 x 200	0.0130	0.0170	0.0187	0.0086	0.0163	0.0170	0.0059	0.0177	0.0148
200 x 200	-0.0132	-0.0120	0.0036	-0.0043	-0.0017	0.0101	-0.0039	0.0024	0.0046
200 x 132	-0.0219	-0.0154	-0.0166	-0.0089	-0.0120	-0.0192	-0.0124	-0.0121	-0.0206
640 x 480	-0.0067	0.0019	0.0115	0.0001	-0.0003	0.0054	0.0070	-0.0022	-0.0012
800 x 600	-0.0614	0.0630	-0.0365	-0.0621	0.0647	-0.0377	-0.0622	0.0644	-0.0272
640 x 480	-0.0067	-0.0036	-0.0081	-0.0065	-0.0046	-0.0102	-0.0089	-0.0055	-0.0117
200 x 200	-0.0001	-0.0099	-0.0037	-0.0031	-0.0106	-0.0057	0.0029	0.0040	0.0032
900 x 600	0.0007	0.0212	0.0123	0.0959	0.0119	0.0080	0.0932	0.0054	0.0050
200 x 133	0.0004	-0.0313	-0.0252	-0.0008	-0.0236	-0.0287	-0.0030	-0.0277	-0.0235
200 x 150	-0.0080	0.0034	-0.0052	-0.0026	0.0052	-0.0002	-0.0090	0.0030	0.0004

3.1.3 Correlation Analysis of Adjacent Pixels

We have also analyzed the correlation between two vertically and horizontally adjacent pixels in about fifty plain images and their corresponding encrypted images. In Fig 4, we have shown the distributions of horizontally adjacent pixels of red, green and blue channels in the image 'Lena' and their corresponding encrypted image. Particularly, in Frames (a), (b) and (c), we have depicted the distributions of two horizontally adjacent pixels of red, green and blue channels respectively in the plain image (Fig 3(a)). Similarly in Frames (d), (e) and (f) respectively, the distributions of two horizontally adjacent pixels in its corresponding encrypted image (Fig 3(e)) have been depicted. Similarly, in Fig 5, we have shown the distributions of vertically adjacent pixels of red, green and blue channels in the plain image 'Lena' and its corresponding encrypted image. We observe from correlation charts and Table 4 that there is a negligible correlation between the two adjacent pixels in the ciphered image. However, the two adjacent pixels in the original image are strongly correlated. Correlation in the encrypted images is very small or negligible when the proposed encryption scheme is used. Hence the proposed scheme has good permutation and substitution properties.

Table 4. Correlation coefficient for two adjacent pixels in the original and its encrypted image

	Original image (Fig 3a)	Encrypted image (Fig 3e)
Horizontal	0.8710	0.0083
Vertical	0.4668	-0.0162
Diagonal	0.6737	0.0078

3.1.4 Information Entropy

Illegibility and indeterminateness are the main goals of image encryption. This indeterminateness can be reflected by one of the most commonly used theoretical measure - information entropy. Information entropy expresses the degree of uncertainties in the system and express by (6).

$$H(m) = - \sum_{i=0}^{2^N-1} P(m_i) \log_2 [P(m_i)] \tag{6}$$

where $P(m_i)$ is the emergence probability of m_i . If every symbol has an equal probability, i.e., $m = \{m_0, m_1, m_2, \dots, m_{2^8-1}\}$ and $P(m_i) = 1/2^8 (i=0, 1, \dots, 255)$, then the entropy is $H(m) = 8$ which corresponds to an ideal case. Practically, the information entropies of encrypted images are less compared to the ideal case. To design a good image encryption scheme, the entropy of encrypted image close to the ideal case is expected.

For the proposed image cipher the information entropy is $H(m) = 7.99$, which is very close to the ideal value. This means a high permutation and substitution is achieved by the proposed algorithm. Proposed algorithm has a robust performance against the entropy attack.

Table 5. Entropy values for different images

Images	Entropy of plain images	Entropy of encrypted images
Lena	7.7502	7.9996
Baboon	7.6430	7.9979
Peppers	7.7150	7.9984
Tiger	7.8261	7.9991
Bear	7.5870	7.9973

3.2 Key Sensitivity Analysis

An ideal image cipher should be extremely sensitive with respect to the secret key used in the algorithm. Flipping of a single bit in the secret key, it should produce a widely different ciphered image. This guarantees the security of a cryptosystem against brute-force attacks to some extent. We have tested the sensitivity with respect to a tiny change in the secret key for several images. One example for plain image ‘Lena’ is discussed below:

- a. Plain image (Fig 3(a)) is encrypted by using the secret key ‘D6DA750B4C1F78D328EA25E6B15CF9E4’ and the resultant encrypted image is referred as image Fig 6(a).
- b. The encrypted image (Fig 6(a)) is decrypted by making a slight modification in the original key ‘E6DA750B4C1F 78D328EA25E6B15CF9E4’ and the resultant decrypted image is referred as image Fig 6(b).
- c. The encrypted image (Fig 6(a)) is decrypted by making a slight modification in the original key ‘D6DA750B4C1 F78D328EA25E6B15CF9E3’ and the resultant decrypted image is referred as image Fig 6(c).
- d. The encrypted image (Fig 6(a)) is decrypted by making a slight modification in the original key ‘D6DA750B4C1F 78D428EA25E6B15CF9E4’ and the resultant decrypted image is referred as image Fig 6(d).

With a small change in the key at any position, one is not able to recover the original image. It is not easy to compare decrypted images with naked eyes. To compare decrypted images, we have calculated the correlation coefficient between encrypted images and various decrypted images and results are given in Table 6. The correlation coefficients are negligible.

Having the right pair of secret key is an important part while decrypting the image, as a similar secret key (with one bit change) will not retrieve the exact original image. Above example shows the effectiveness of the proposed technique as the decryption with a slightly different secret key does not reveal any information to an unauthorized person.

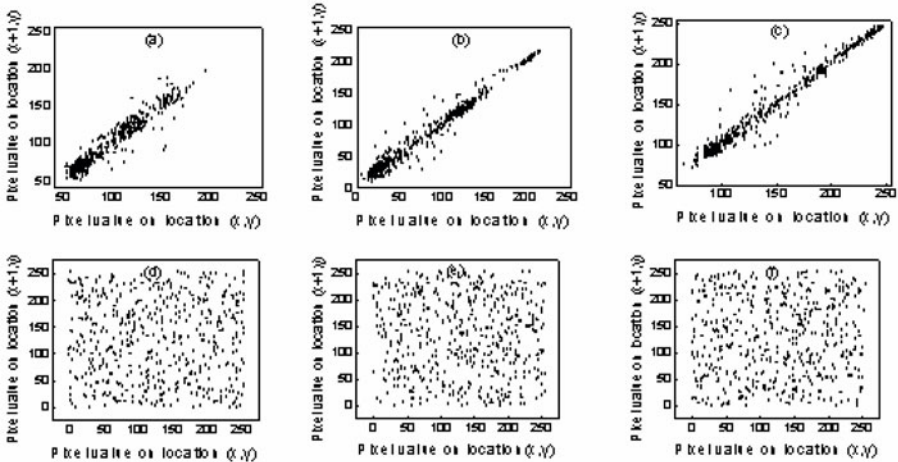


Fig. 4. Distributions of horizontally adjacent pixels of RGB channels in the plain image ‘Lena’ and its encrypted image

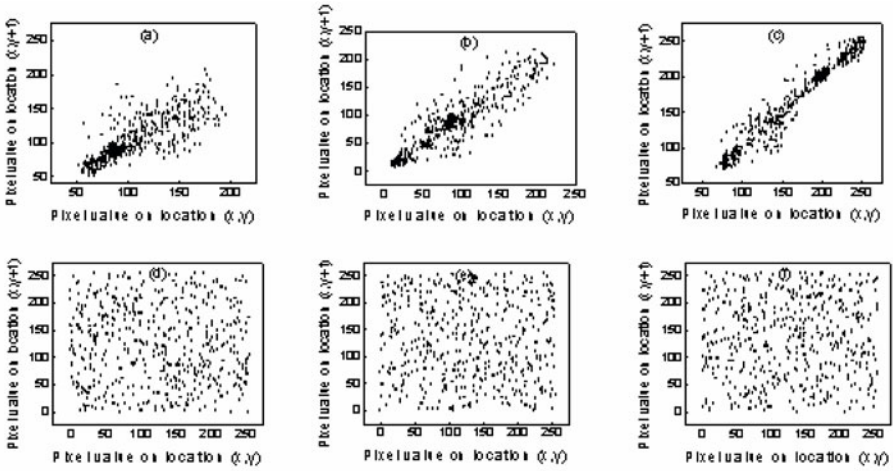


Fig. 5. Distributions of vertically adjacent pixels of RGB channels in the plain image ‘Lena’ and its encrypted image

Table 6. Correlation coefficient between RGB channels of different decrypted images

Images	Correlation coefficient
Fig 6(a) and Fig 6(b)	$C_{RR}=0.0182, C_{GG}=0.0290, C_{BB}=0.0162$
Fig 6(a) and Fig 6(c)	$C_{RR}=0.0105, C_{GG}=0.0089, C_{BB}=0.0103$
Fig 6(a) and Fig 6(d)	$C_{RR}=0.0284, C_{GG}=0.0414, C_{BB}=0.0255$

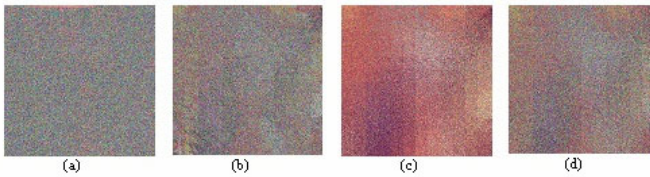


Fig. 6. Decrypted images corresponding to image ‘Lena’ with slightly different secret keys

3.3 Plain Image Sensitivity

Implementation of various types of attacks such as known plain text attack, chosen plaintext attack and more advanced adaptive chosen plaintext attack is done by making a slight change in plain image and compare the pair of encrypted images in order to extract some relationship between plain image and encrypted image, which helps attackers in determining the secret key. Such analysis is termed as differential cryptanalysis in cryptographic terminology. If one minor change in the plain image causes large changes in the encrypted image then such differential analysis may become

useless. Thus, much difference between encrypted forms is expected in order to keep high security. Plain image sensitivity is the mostly analyzed by the number of pixel change rate (NPCR). NPCR means the number of pixels changed in the encrypted image when only one pixel value is changed in plain image. For a larger value of NPCR, plain image has the higher sensitivity.

In our plain image sensitivity test, we changed pixel at position (1,3) from (62,32,96) to (255,255,255) in plain image 'Lena'. Table 7 shows the values of NPCR for each rounds of permutation and substitution process which is over 97%. Hence the proposed encryption scheme is resistant against differential attacks.

3.4 Image Quality Criterion

A very useful measure of the performance of the decryption procedure is the mean square error (MSE). The smaller MSE value, the better the image quality recovered. On the contrary, the greater the MSE value, the worse the image quality recovered. For P and P' being a plain image and the decrypted image respectively, the MSE for the each color component (RGB) is defined as

$$MSE = \frac{\sum_{m=1}^M \sum_{n=1}^N [P(m, n) - P'(m, n)]^2}{M \times N} \quad (7)$$

where (m,n) are the pixel coordinates, $M \times N$ is the number of pixels of the images considered. $P(m,n)$ and $P'(m,n)$ are the original and decrypted image recovered respectively. We have calculated MSEs for the red, green and blue components of the decrypted images with respect to their original for around fifty images. In all test cases, it was found to be approximately zero. Hence, proposed image cipher comes under the category of lossless image cipher.

3.5 Key Space Analysis

The key space of the proposed image cipher is large enough to resist all kinds of brute force attacks. The experimental results also demonstrate that our scheme is very sensitive to the secret key. If the decryption key changes slightly, the decrypted image will be greatly different from the original plain image as shown in Fig 6. Secret key used in the image cipher should be neither too long nor too short. A larger secret key decreases the encryption speed and is not preferred for real time image transmission whereas a choice of smaller secret key results in an easy cryptanalysis. In the proposed encryption scheme, a secret key of 128-bits long is used. Thus, it has 2^{128} different combinations (3.40×10^{38}). An image cipher with such a large key space is sufficient for resisting various brute-force attacks. It is possible to increase the number of bits for key in hardware implementation. However, by increasing the key length, volume of hardware is increased and consequently speed of the system is decreased. With respect to the speed of the today's computers, the key space size should be more than 2^{100} , in order to avoid brute-force attacks [1].

Table 7. Number of pixel change rate (NPCR)

NPCR		Rounds in permutation process							
		1	2	3	4	5	6	7	8
Rounds in substitution process	1	97.71	97.87	98.12	98.89	98.79	99.12	99.23	99.20
	2	97.80	97.82	98.35	98.84	98.97	99.16	99.34	99.30
	3	97.89	97.85	97.99	98.66	98.89	99.04	99.38	99.42
	4	98.20	98.28	98.65	98.66	98.73	98.82	99.20	99.39
	5	98.67	98.71	98.69	98.72	98.78	98.94	98.99	99.33
	6	98.98	98.95	98.96	98.90	99.02	99.10	99.12	99.20
	7	99.23	99.20	99.24	99.20	99.11	99.16	99.20	99.42
	8	99.29	99.20	99.15	99.30	99.31	99.33	99.26	99.46

3.6 Time Analysis

Apart from the security consideration, encryption/ decryption rate of the algorithm is also an important aspect for a good image cipher. We have also measured time taken by the proposed cipher to encrypt/decrypt various different sized colour images. The time analysis has been done on a personal computer with Intel core 2 duo 1.8Ghz processor and 1.5GB RAM. The results are summarized in Table 8, which clearly predicts an average encryption rate of proposed scheme is 380KB/second.

Table 8. Encryption rate of proposed image cipher

Image dimen-	Image size	Average time
512x512	768 KB	2.26s
200x200	117 KB	0.27s
200x305	178 KB	0.44s
800x600	1.37 MB	4.17s

4 Conclusion

We propose a new non-chaos based image encryption scheme using a secret key of 128-bit size. In the algorithm, image is partitioned into several key based dynamic blocks and each block is passed through the eight rounds of permutation as well as substitution process. In permutation process, sequences of block pixels are rearranged within itself by a zigzag approach whereas block pixels are replaced with another by using difference transformation in substitution process. We have carried out an extensive security and performance analysis of the proposed image encryption technique using various statistical analysis, key sensitivity analysis, differential analysis, key space analysis, speed performance, etc. Based on the results of our analysis, we conclude that the proposed image encryption technique is perfectly suitable for the secure image storing and transmission.

Acknowledgments. One of us (VP) acknowledges to the Science and Engineering Research Council (SERC), Department of Science and Technology (DST), Government of India for the Fast Track Young Scientist Research Grant (SR/FTP/PS-17/2009).

References

1. Yun, L., Feng-Ying, H.: New image encryption algorithm based on combined high – dimension chaotic system. *Computer Engineering and Applications* 45(1), 103–107 (2009)
2. Çokal, C., Solak, E.: Cryptanalysis of a chaos-based image encryption algorithm. *Physics Letters A* 373(15), 1357–1360 (2009)
3. Guan, Z.H., Huang, F., Guan, W.: Chaos-based image encryption algorithm. *Physics Letter A* 346(1-3), 153–157 (2005)
4. Jia, C., Huai-Xun, Z., Jian-Yang, Z.: Image encryption algorithm based on composite discrete chaotic system. *Computer Engineering* 35(6), 162–163 (2009)
5. Yi-Xiao, S., Zhuan-Cheng, Z., Jia-Chen, W., Bo, Q.: Image chaotic encryption algorithm based on sampling without replacement. *Computer Engineering* 35(5), 139–141 (2009)
6. Wei-bin, C., Xin, Z.: Image encryption algorithm based on Henon chaotic system. *International conference on Image Analysis and Signal Processing (IASP)*, pp. 94–97 (2009)
7. Rhouma, R., Meherzi, S., Belghith, S.: OCML-based colour image encryption. *Chaos, Solitons and Fractals* 40(1), 309–318 (2009)
8. Yang, H., Liao, X., Wong, K.-W., Zhang, W., Wei, P.: A new cryptosystem based on chaotic map and operations algebraic. *Chaos, Solitons and Fractals* 40(5), 2520–2531 (2009)
9. Xiao, D., Liao, X., Wei, P.: Analysis and improvement of a chaos-based image encryption algorithm. *Chaos, Solitons and Fractals* 40(5), 2191–2199 (2009)
10. Wang, Y., Wong, K.-W., Liao, X., Xiang, T.: A block cipher with dynamic S-boxes based on tent map. *Communications in Nonlinear Science and Numerical Simulation* 14(7), 3089–3099 (2009)
11. Patidar, V., Pareek, N.K., Sud, K.K.: A new substitution–diffusion based image cipher using chaotic standard and logistic maps. *Communications in Nonlinear Science and Numerical Simulation* 14(7), 3056–3075 (2009)
12. Rhouma, R., Ercan, S., Safya, B.: Cryptanalysis of a new substitution–diffusion based image cipher. *Communications in Nonlinear Science and Numerical Simulation* 15(7), 1887–1892 (2009)
13. Tong, X., Cui, M.: Image encryption scheme based on 3D baker with dynamical compound chaotic sequence cipher generator. *Signal Processing* 89(4), 480–491 (2009)
14. Li, C., Li, S., Chen, G., Halang, W.A.: Cryptanalysis of an image encryption scheme based on a compound chaotic sequence. *Image and Vision Computing* 27(8), 1035–1039 (2009)
15. Wang, Y., Wong, K.-W., Liao, X., Xiang, T.: A chaos-based image encryption algorithm with variable control parameters. *Chaos, Solitons and Fractals* 41(4), 1773–1783 (2009)
16. Wong, K.-W., Kwok, B.S.-H., Yuen, C.-H.: An efficient diffusion approach for chaos-based image encryption. *Chaos, Solitons and Fractals* 41(5), 2652–2663 (2009)
17. Arroyo, D., Li, C., Li, S., Alvarez, G., Halang, W.A.: Cryptanalysis of an image encryption scheme based on a new total shuffling algorithm. *Chaos, Solitons and Fractals* 41(5), 2613–2616 (2009)
18. Gao, T.G., Chen, Z.Q.: Image encryption based on a new total shuffling algorithm. *Chaos, Solitons and Fractals* 38(1), 213–220 (2008)

19. Tang, Y., Wang, Z., Fang, J.: Image encryption using chaotic coupled map lattices with time-varying delays. *Communications in Nonlinear Science and Numerical Simulation* (2009) (in press)
20. Wang, X.-y., Chen, F., Wang, T.: A new compound mode of confusion and diffusion for block encryption of image based on chaos. *Communications in Nonlinear Science and Numerical Simulation* 15(9), 2479–2485 (2009)
21. Alvarez, G., Li, S.: Some basic cryptographic requirements for chaos-based cryptosystems. *International Journal of Bifurcation and Chaos* 16(8), 2129–2151 (2006)
22. Gao, H., Zhang, Y., Liang, S., Li, D.: A new chaotic algorithm for image encryption. *Chaos, Solitons and Fractals* 29, 393–399 (2006)
23. Patidar, V., Pareek, N.K., Sud, K.K.: Modified substitution–diffusion image cipher using chaotic standard and logistic maps. *Communications in Nonlinear Science and Numerical Simulation* 15(10), 2755–2765 (2009)
24. Mazloom, S., Eftekhari-Moghadam, A.M.: Color image encryption based on coupled nonlinear chaotic map. *Chaos, Solitons and Fractals* 42(3), 1745–1754 (2009)
25. Wang, Y., Wong, K.-W., Liao, X., Chen, G.: A new chaos-based fast image encryption algorithm. *Applied Soft Computing* (2009) (in press)
26. Yong-Liang, X., Li-Min, X.: An Image encryption approach using a shuffling map. *Communication Theoretical Physics* 52, 876–880 (2009)
27. Shannon, C.E.: Communication theory of secrecy systems. *Bell Systems Technical Journal* 28, 656–715 (1940)

Deciphering the Main Research Themes of Software Validation – A Bibliographical Study

Tsung Teng Chen¹, Yaw Han Chiu², and Yen Ping Chi²

¹ Institute of Information Management, National Taipei University, Taipei, Taiwan

² Department of Management Information Systems, National Chengchi University, Taipei, Taiwan

timchen.ntpu@msa.hinet.net, 94356511@nccu.edu.tw,

ypchi@mis.nccu.edu.tw

Abstract. Software validation is an important issue since software is embedded in various devices and these devices are ubiquitous in our daily life nowadays. Software validation is an attempt to ensure the software product fulfills its specific intended purpose. As software is getting more complex, the software validation process is getting more complicated and time consuming. The study of software validation is an important active research field with voluminous publications. The sheer volume of software validation related literatures hinders our comprehending of its content and context. We therefore utilize the intellectual structure construction and knowledge domain visualization techniques developed by the information scientist to ease the task of understanding the main themes of this important research domain. Base on our analysis, we can see the research themes of software validation may be divided into three main groups: the first one deals with systems with interweaving hardware and software related issues, such as real-time systems, parallel systems; the second one deals with model checking related issues, including automata theory and temporal logic; the third one deals the modeling and simulation of concurrent and distributed systems.

Keywords: Software Validation, Intellectual Structure, Domain Visualization, Citation Analysis.

1 Introduction

In some large software projects, there are over one million source codes, and it is difficult to valid such a complex software. Facing such a daunting situation, how do we effectively and efficiently validate a complicated software is an intriguing issue. Software developed for the aviation, automobile, mass transportation, medical, and other life connected industries are very critical. A system malfunction or failure may cause huge economic loss or even life threatening. Therefore, it is a very critical task for us to verify and validate software reliably.

Since the literature relevant to software validation is voluminous, we need to utilize the methodology devised by information scientists that has been successfully

applied in analyzing large collections of literature. We applied the intellectual structure building method to the study of software validation that reveals the major research themes and their inter-relationships. However, effective and efficient visualization of such a vast amount of knowledge is a challenging task. We therefore utilize the intellectual structure and knowledge domain visualization techniques to ease the task of understanding this vast research domain.

2 Literature Review

2.1 Software Validation

Software Validation is generally called Verification and Validation (V & V). The purpose is to ensure that a system conforms to its specifications and expectations of the customers [1]. The Verification and Validation process areas are related and similar. Therefore, they have often been confused by people, but they do address different issues. Capability Maturity Model Integration (CMMI) has the following definitions [2]:

Verification: It ensures that you are building a product according to its requirements, specifications, and standards.

Validation: It ensures that your product will be usable once it is in its intended environment.

In short, Verification makes sure that the software conforms to its specification and was rightly built, and Validation assures that the software does what the user really requires and the right product was developed.

2.2 Citation Analysis

The citation analysis method was pioneered by Garfield et al [3]. It is one of the bibliographical methods, and mainly applied in analyzing the citation relationships between documents. It is an analytic tool that can be used to reveal the latent information from voluminous literatures [4]. It has become the dominant method for the empirical study of the structures of scientific communications, and has been applied to facilitate the understanding of the research themes and trends. The concept of co-citation was proposed in 1973 by Small, who developed the citation map as an analytical tool for interpreting the results of the literature analysis [5].

The citing and cited relation between the articles of a related field may form a network structure. These networked relationships may be abstracted into a graph or a citation network. A citation network reveals the citation relationships between the articles, and may reveal the important nodes (article) as well. In Fig. 1, two citation relationships are exhibited: the direct citation and the co-citation relationship. Article D refers to article A and B at the same time, then there are direct citation relationships between the pairs (D, A) and (D, B). There is a co-citation relationship between (A, B), which dues to article D refer both A and B article.

If two or more articles are both cited by another article, then these two cited articles are said to be co-cited. The number of citation count of article A, B and C are 1, 2, and 1, respectively. Article A and B is co-cited once, the number of co-citation count of

article A and B is both 1. Similarly, B and C are also co-cited once. We can build the adjacent matrix A to represent the relationship of direct citation, and the co-citation matrix can be derived by $A^T A$ (the transpose matrix of A multiplies matrix A). The co-citation counts between the articles are used as a relativity measurement [6].

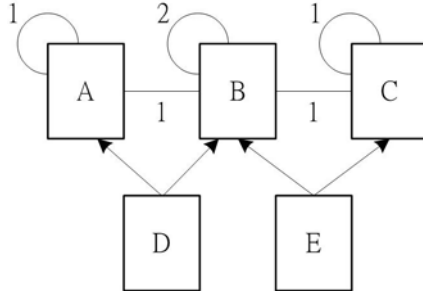


Fig. 1. A citation network whereas the citation count of B is 2, and the citation count of A and C is 1; the co-citation count of A and B, B and C are both 1

2.3 Knowledge Domain Visualization

Comprehending and following the development of the scientific knowledge through literature tracking is one of the crucial research activities. Today, the scientific literature database store vast amounts of human knowledge in the text format, but how to efficiently and effectively explore this vast information is a daunting task. Visual data exploration may provide better insight by visualizing the data, and help solve this problem [7].

Domain visualization is aiming to provide users with a wider range of ways to explore scientific literature. Using visual tools to explore the results, the complex information may be understood more quickly. A picture is worth a thousand words, so we use the intellectual structure map [8, 9] to exhibit the large amount of information of the software validation domain. Domain visualization not only enables researchers and learners to explore and follow up the development of the knowledge in a specific domain, but also provides an interface that can directly interact with the data. Knowledge Domain Visualization [10] aims to reveal the most significant intellectual structure associated with a subject domain [11].

3 Research Methodology

3.1 Data Collection and Reduction

Our study uses the term "Software Validation" to query the CiteSeer [12] mirror site in Singapore in September 2008 and retrieve 1,787 seed papers. Following the forward citing and backward cited relationships of these seed papers, we collected 327,148 papers, which contain 992,323 citation arcs. In order to keep the papers

which are more relevant to the query terms, we take only the papers that are two links away from these seed papers, e.g., if the paper A refers to paper B and paper B refers to paper C, then paper A and C is said to be two links away. The resulted dataset contains 52,592 papers that are two links away from the seed papers, and there are 179,670 citation arcs between these papers. We further pruned out papers that were cited less than 10 times, which left 2,338 papers and 15,972 citation arcs between them. We then compute the co-citation matrix from the citation adjacent matrix and filter out papers that are co-cited less than six times. The remaining 176 papers, which have 457 weighted co-citation edges among them, are used for further analysis.

3.2 Intellectual Structure Construction and Visualization Process

The remaining steps of the procedure for the intellectual structure building and visualization process are listed as following:

1. Applying factor analysis using the co-citation matrix discussed in the previous section, and derives the Pearson correlation coefficient matrix. The top 20 factors are selected to represent the main research themes.
2. Using the Pathfinder [13] scaling algorithm to keep only the strongest links in the relateness graph derived from the Pearson correlation coefficient matrix, a pathfinder network (PFNET) graph is generated.
3. Applying the minimum spanning tree algorithm to the PFNET graph to eliminate the redundant edges.
4. Drawing the graph derive from step 3.

4 Deciphering and Interpretation of the Main Research Themes of Software Validation

4.1 Decipher the Main Research Themes Using Factor Analysis

We extract the representative research themes from the top 20 factors obtained from the Factor Analysis. The main purposes of the factor analysis are: (1) to reduce the number of variables and (2) to detect structure in the relationships between variables. Therefore, factor analysis is applied as a data reduction or structure detection method [14]. We use factor analysis to combine correlated variables (papers) into one component (research theme). The co-citation matrix is the input of the factor analysis. Table 1 shows the research theme represented by each factor. The total explained variance is approximately 78.84%, which is considered high enough for an exploratory study. Since a paper may be assigned to more than one factor, the factor loading of a paper is used to uniquely ascribe a paper to one factor. Papers with a factor loading 0.6 or above are collected and studied to decide the content of the factor. The research themes represented by each factor are deciphered and listed in Table 1.

Table 1. Research Themes Represented by Factors

Factor	Research Theme	Explained Variance (%)	Total Explained Variance (%)
1	The Modeling, Specifying, and Proving of Real-time Systems	10.791	10.791
2	Specification and Analysis of Hybrid Systems	10.714	21.505
3	Program Slicing and System Dependency Graph	8.962	30.466
4	Analysis of Concurrent Systems	5.881	36.347
5	Efficient Manipulation of Boolean Function	5.792	42.139
6	Temporal Logic	5.402	47.541
7	Alias Analysis	5.261	52.802
8	Mutual Exclusion Problem	3.187	55.989
9	Model Checking	3.024	59.012
10	Program Verification Using Binary Decision Diagrams	2.763	61.775
11	Requirement Specification and Engineering	2.339	64.114
12	Mu-Calculus based Model Checking	2.086	66.200
13	State Space Reduction	1.911	68.112
14	Communication and Concurrency	1.882	69.993
15	Simulation Program	1.853	71.847
16	Computational Logic	1.610	73.456
17	Timed Input-Output Automation	1.408	74.865
18	Reactive Systems	1.383	76.248
19	Code Optimization	1.353	77.601
20	Null*	1.234	78.835

*When there is no paper under a factor with loading over 0.6, we are unable to interpret the content of a factor through papers ascribed to it.

4.2 The Main Research Themes of the Study of Software Validation

The descriptive research themes are given by reading and summarizing the papers in each theme.

- **Theme 1** - The Modeling, Specifying, and Proving of Real-time Systems

A real-time system has to meet some critical time constraints imposed by its operational conditions. The time dimension of a real-time system, in addition to other requirements, makes the validation of such a system a difficult task. An effective and robust modeling and specificative formalism for the real-time system serve as the foundation for its correctness and validation. The popular Propositional Temporal Logic (PTL), whose mathematical model is a one-way infinite state sequence, is

widely used in the specification and verification of the non real-time system. However, PTL lacks the clock's concept, which a real-time system has to consider, cannot be applied in the analysis of real-time systems. Alur and Henzinger proposed the timed PTL (TPTL), which addressed the limitation of PTL, provide a robust formalism for specifying the real-time system [15].

- **Theme 2 - Specification and Analysis of Hybrid Systems**

A hybrid system is a dynamic system that exhibits both continuous and discrete dynamic behavior. Hybrid systems have become a major research topic in control engineering and other related disciplines. Many different models have been proposed for describing them. Since a hybrid system exhibits dual characteristics, the conventional analysis technique, which is suitable for either continuous or discrete system, is not applicable in the specification and analysis of a hybrid system. The development of effective algorithmic analysis methods for the verification of hybrid systems is the main research theme.

- **Theme 3 - Program Slicing and System Dependency Graph**

Locating all statements in a program that directly or indirectly affect the value of a variable occurrence is referred to as Program Slicing. These located statements constitute a slice of the program with respect to the variable occurrence. Program slicing was defined by Mark Weiser, which is useful in program debugging, maintenance, and other applications area such as understanding program behavior [16]. Program slices may be efficiently computable by the reachability analysis using the program's System Dependency Graph (SDG). Program slicing and SDG may greatly expedite the software testing activities, which are required in software validation.

- **Theme 4 - Analysis of Concurrent Systems**

Large software systems are often organized as collections of cooperating concurrent processes. These systems are hard to analyze, especially when the non-determinacy is incorporated. The articles in this theme discuss the tools and methods for analyzing and describing the concurrent systems.

- **Theme 5 - Efficient Manipulation of Boolean Function**

Efficient algorithms to perform state enumeration for finite state machine can be applied to a wide range of problems, which include implementation verification, design verification, sequential testing, and sequential synthesis. Traditional state enumeration methods cannot handle machines with more than a few million states. To remedy this limitation, studies in this theme develop methods to represent automata's behavior and state space, which simplifies the situation and reduces the required computational resources for huge state machines.

- **Theme 6 - Temporal Logic**

The term Temporal Logic has been broadly used to cover all approaches to the representation of temporal information within a logical framework, and also more narrowly to refer specifically to the modal-logic type of approach introduced around 1960 by

Arthur Prior under the name of Tense Logic. Applications of Temporal Logic include its use as formalism for clarifying philosophical issues about time, as a framework within which to define the semantics of temporal expressions in natural language and as a tool for handling the temporal aspects of the execution of computer programs. Articles in this theme discuss the verification of a concurrent program, which may be expressed in temporal logic formalism.

- **Theme 7 - Alias Analysis**

Aliasing occurs when two or more variables refer the same storage location in a program. Alias analysis is used to detect the aliasing situation, which may cause the compiler to generate inefficient code.

- **Theme 8 - Mutual Exclusion Problem**

In 1965, the mutual exclusion problem in concurrent programming was defined by Dijkstra [17], which is characterized by the atomic read and write operations of a shared memory. This problem may be illustrated by two or more processes competing for some shared resources, such as a global variable. Many solutions have been published to guarantee the mutual exclusive access to a critical section among the competing processes.

- **Theme 9 - Model Checking**

Model checking is a formal method for verifying the temporal logic properties of the finite state systems, which allows developers to detect errors automatically. To use this technique, the finite state model (M) of a system and a specification formula (F) is given for the model checker to test if the model conforms to the specification. M is an automata and the F is the temporal logic. M and F serve as inputs to a model checker. The studies in this theme are mainly applying model checking in analyzing the concurrent systems to check if there are deadlocks or infinite loops.

- **Theme 10 - Program Verification Using Binary Decision Diagrams**

A binary decision diagram (BDD) is a directed acyclic graph which compactly represents a Boolean function with lesser memory. Therefore, it can efficiently represent the system's state information to facilitate the analysis of a program. BDD forms a canonical representation, making testing of functional properties such as satisfiability and equivalence straightforward.

- **Theme 11 - Requirement Specification and Engineering**

The study of requirement specification concerns new techniques to define precise, concise, and unambiguous requirement specifications, with which the completeness and consistency are easy to validate. The studies of requirement engineering discuss new requirement engineering methods, such as CoRE (Consortium Requirements Engineering), which has been applied in many cases, e.g., Lockheed applied the CoRE method in the C-130J software requirements' engineering process [18]. It has also been applied in the requirements' development of the flight software for the A-7 aircraft [19].

- **Theme 12** - Mu-Calculus based Model Checking

The mu-calculus may be thought as extending the computation tree logic (CTL) with a least point (μ) and greatest point (ν) operator [20], which makes it possible to give extreme fix point characterizations of the correctness properties. Many temporal logics can be translated into the mu-calculus formalisms, such as CTL and linear temporal logic (LTL) etc. It permits characterization of many correctness properties for parallel and reactive systems in terms of the external fix points of predicate transformers.

- **Theme 13** - State Space Reduction

The state explosion is the most critical problem facing Model Checking methods. The state explosion problem is characterized by the exponentially grown system states as the number of components in the system increased. The study in this theme presents a simple method, which applies model checking without incurring most of the cost of the concurrency modeling. It exploits the symmetric characteristics in concurrent systems, which makes it avoid or moderate the state explosion problem.

- **Theme 14** - Communication and Concurrency

This theme studies the communication and concurrency issues of systems. The communicating sequential processes (CSP), published by C.A.R Hoare [21], is a language for describing patterns of interaction. CSP is a specification language for the concurrent and distributed systems, and it has been applied as a tool for specifying and verifying the concurrent aspects of a diversity of systems.

- **Theme 15** - Simulation Program

The simulation abstracts from a program such details as how the sequencing is controlled and how data is represented. This makes the model mimicking the real systems' response to the temporal events, and facilitates the evaluation and improvement of the system performance.

- **Theme 16** - Computational Logic

A computational logic (ACL) is an interdisciplinary field having its roots in artificial intelligence, computer science, logic, and applied mathematics. The automatic inference has influenced the development of artificial intelligence and computer science. ACL provides a rather accurate description of many of the heuristics. This theme mainly introduces a heuristic approach in theorem-proving. It has been successfully applied to the fields of hardware and software verification.

- **Theme 17** – Timed Input-Output Automation

The input-output automaton (IOA) model was developed by Lynch and Tuttle for the modeling of an asynchronous distributed network. It distinguishes between those actions whose performance is under the control of the automaton and these other actions which are affected by its environment. Tuttle et al. presents the timed input-output automaton (TIOA) model based on IOA, which is applicable in analyzing in real-time systems.

- **Theme 18 - Reactive Systems**

A reactive system is a system that changes its actions in response to stimuli from within, or from its operating environment. Therefore, the uncertainty from the environment makes it more difficult to analyze and forecast the behavior of a reactive system. Lustre and Esterel are synchronous programming languages that are designed to programming complex reactive system, and they have also been used to describing hardware such as the automatic control and monitoring systems.

- **Theme 19 - Code Optimization**

Constant propagation analysis is one of the main methods used in codes optimization. Constant propagation is a code improvement technique, which the compiler infers the value a variable may assume at the run time. If the compiler can establish that a variable will always have a known constant value in run time, it may replace the variable by the constant itself to optimize the generated object codes.

4.3 Visualizing the Intellectual Structure of the Software Validation Field

We show the Intellectual Structure map of the research field of Software Validation in Fig. 2. An article is represented by a node in the map. The number in the parenthesis is the factor (research theme) number which the article belongs to; articles under the same factor are painted with the same color. The distance between articles is a function of the Pearson correlation coefficient, which places the less correlated articles apart and highly correlated articles spatially adjacent. From Fig. 2, we can see there are three clusters, which are described as follows.

Group 1 is the biggest cluster, which consists of papers from factors 1, 2, 6, 9, and 17. The modeling, specifying, and proving of real-time Systems (theme 1) is resided in the center of this cluster, and it connects with the specification and analysis of hybrid systems (theme 2), Temporal Logic (theme 6), and Timed Input-Output Automation (theme 17). It implies that the research of the real-time system modeling provides the theoretical basis for these studies.

Group 2 includes Model Checking (factor 9) as the central cluster, and it connects with the Analysis of Concurrent Systems (factor 4) and a portion of Program Verification using Binary Decision Diagram (factor 10). Referring Fig. 2, we postulate the Model Checking study provides fundamental theories for the analysis of concurrent systems and program verification.

Group 3 includes Communication and Concurrency (factor 14) and Simulation Program (factor 15). Simulation is utilized to simulate distributed systems and internet communication to harness the complexities of the temporal event.

In summarization, software validation utilizes model checking to verify the hardware and software systems. The study of model checking encompasses many fields; it roots in mathematics algebra and includes knowledge from many fields such as the temporal logic and automata theory. We can see the indirect relationships between some research themes through the intellectual structure map.

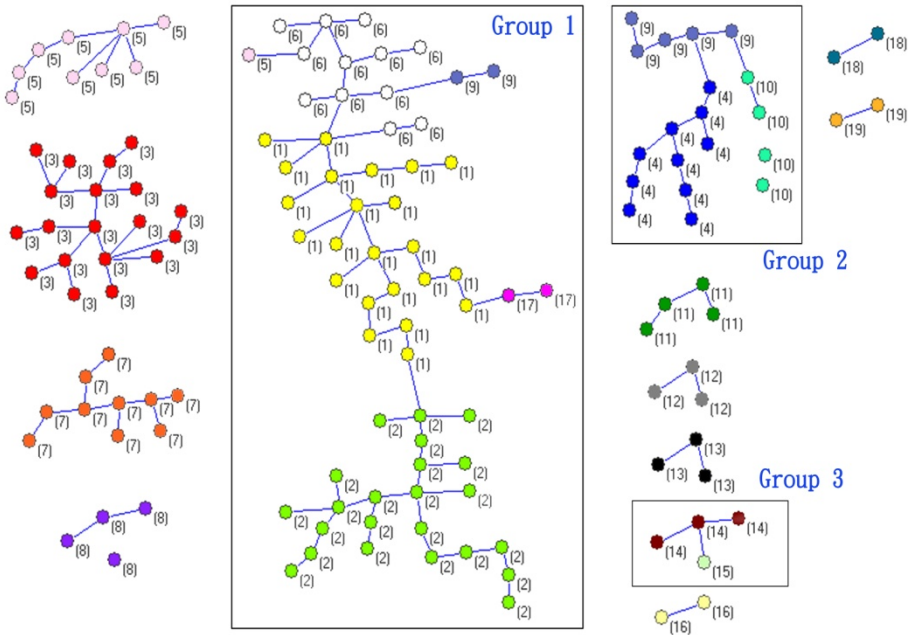


Fig. 2. The Intellectual Structure map of the studies of software validation, whereas papers are represented by colored nodes; papers belong to the same theme are painted in the same color; the number in the parenthesis corresponds to the factor numbers in Table 1

5 Conclusion

We deciphered the main themes and constructed the intellectual structure map for the software validation studies. From the structure map, we can see that the studies of modeling, specifying, and proving the correctness of real-time systems provide one of the theoretical foundations for the software validation study. Since the verification and the validation of real-time systems is a very strenuous task, it is logical that considerable formalisms and theories have been developed to help us to cope with this complicated problem. We can see that the research field of the software validation study includes specification and modeling formalisms and tools for these complex systems such as real-time systems, hybrid systems, concurrent systems, and reactive systems. This development may be due to the validation methodologies and tools that are applicable to conventional data processing systems are inadequate to tackle these complex systems. Besides the theories and formalisms for the specification and modeling of complex systems, the other main areas in this field include the time and space reduction techniques, which are applicable in the specification and modeling of the complex systems.

We try to provide a broader view to the study of software validation through the construction and interpretation of its intellectual structure. As we can see from the intellectual structure map, we can uncover the main research themes – techniques for the specification and modeling of complex systems; important supplemental studies – time

and space reduction methods; frequent application areas – applied in real-time systems, hybrid systems, concurrent systems, and reactive systems; and other relevant studies such as program optimization.

References

1. Sommerville, I.: Software Engineering, 8th edn. Addison-Wesley, Reading (2007)
2. Team C. P.: CMMI for Development, version 1.2, CMMI-DEV v1. 2. Technical Report SEI-2006-TR-008 (2006)
3. Garfield, E., Sher, I., Torpie, R.: The Use of Citation Data in Writing the History of Science. Institute for Scientific Information (1964)
4. Chen, T., Xie, L.: Identifying Critical Focuses in Research Domains. 9th International Conference on Information Visualisation - IV 2005, 135–142 (2005)
5. Small, H.: Co-citation in the Scientific Literature: a New Measure of the Relationship between Two Documents. *Journal of the American society for information science* 24, 265–269 (1973)
6. Garfield, E., Malin, M., Small, H.: Citation Data as Science Indicators. In: *Toward a Metric of Science: the Advent of Science Indicators*, New York, pp. 179–207 (1978)
7. Keim, D.A.: Visual Exploration of Large Data Sets. *Communications of the ACM* 44, 38–44 (2001)
8. Chen, T., Yen, D.: Technical Research Themes of the Mobile Ubiquitous Computing. In: *Eighth International Conference on Mobile Business*, pp. 221–226 (2009)
9. Chen, T., Lee, M.: Ubiquitous Computing in Prospect: a Bibliographic Study. In: *Ubiquitous Multimedia Computing*, pp. 57–62 (2008)
10. Borner, K., Chen, C., Boyack, K.: Visualizing Knowledge Domains. *Annual Review of Information Science and Technology* 37, 179–255 (2003)
11. Chen, C.: Domain Visualization for Digital Libraries. In: *IEEE Fourth International Conference on Information Visualisation (IV 2000)*, pp. 261–267 (2000)
12. Giles, C.L., Bollacker, K.D., Lawrence, S.: CiteSeer: An Automatic Citation Indexing System. In: *The Third ACM Conference on Digital Libraries*, pp. 89–98 (1998)
13. Schvaneveldt, R.W.: *Pathfinder associative networks: studies in knowledge organization*. Ablex Publishing Corp. Norwood (1990)
14. Stevens, J.: *Applied Multivariate Statistics for the Social Sciences*. Lawrence Erlbaum Associates, New Jersey (1999)
15. Alur, R., Henzinger, T.A.: A Really Temporal Logic. *J. ACM* 41, 181–203 (1994)
16. Ball, T., Horwitz, S.: Slicing Programs with Arbitrary Control Flow. In: *Adsul, B. (ed.) AADEBUG 1993*. LNCS, vol. 749, pp. 206–222. Springer, Heidelberg (1993)
17. Dijkstra, E.W.: Solution of a Problem in Concurrent Programming Control. *Commun. ACM* 8, 569 (1965)
18. Faulk, S., Finneran, L., Kirby Jr, J., Shah, S., Sutton, J.: Experience Applying the CoRE Method to the Lockheed C-130J Software Requirements. In: *Ninth Annual Conference on Computer Assurance*, Gaithersburg, MD, pp. 3–8 (1994)
19. Heninger, K.: Specifying Software Requirements for Complex Systems: New Techniques and Their Application. *IEEE Transactions on Software Engineering* 6, 2–13 (1980)
20. Emerson, E.: Model-Checking and the Mu-Calculus. *Descriptive Complexity and Finite Models* 31, 185–214 (1997)
21. Hoare, C.A.R.: Communicating Sequential Processes. *Communications of the ACM* 21, 666–677 (1978)

Toward a Comprehension View of Software Product Line

Sami Ouali, Naoufel Kraiem, and Henda Ben Ghezala

RIADI Lab, ENSI, Compus of Manouba
Manouba, Tunisia

samiouali@gmail.com, naoufel.kraiem@ensi.rnu.tn,
henda.bg@cck.rnu.tn

Abstract. Software product line engineering is an approach that develops and maintains families of products while taking advantage of their common aspects and predicted variabilities. Indeed, software product lines (SPL) are an important means for implementing software variability which is the ability of a system to be efficiently extended, changed, customized or configured for use in a particular context. Variability needs in software are constantly increasing because variability moves from mechanics and hardware to software and design decisions are delayed as long as economically feasible. Numerous SPL construction approaches are proposed. Different in nature, these approaches have nevertheless some common disadvantages. We have proceeded to an in-depth analysis of existing approaches for the construction of Software Product Line within a comparison framework in order to identify their drawbacks.

Keywords: Software Product Line, variability, comparison framework.

1 Introduction

The Software Engineering Institute (SEI) defines the software Product Line as [1], “a set of software-intensive systems that share a common, managed set of features satisfying the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way”. Software product line engineering is an approach that develops and maintains families of products while taking advantage of their common aspects and predicted variability’s [2]. The Software product line (SPL) is one of the important means for implementing software variability. Indeed, variability is the ability of a system to be efficiently extended, changed, customized or configured for use in a particular context [3]. Variability needs in software are constantly increasing. Indeed, the variability is moving from mechanics and hardware to software. Currently, design decisions are delayed as long as economically feasible.

SPL engineering is considered as unavoidable approaches to support reuse in systems development, as a viable and important software development paradigm. It allows to companies to realize a real improvements in time to market, cost, productivity, quality and flexibility.

In fact, SPL techniques are explicitly capitalizing on commonality. They try formally to manage the variations of the products in the product line.

In the literature, there are many SPL construction methods [3] [4] [5] [6] [7] [8] [9]... In this article, we present four of these methods. We have tried in our choice to present a diversity of methods which deal with SPL in different way.

Despite their diversity, different SPL construction methods have some common drawbacks. To identify them, we elaborate a comparison framework. From the application of our comparison framework on the four selected methods, we realize that we have a lack of sufficient tool support for them and for their interactivity with their users. Moreover, some of the proposed methods are using proprietary notations which can handle some problems of standardization and interoperability...

Thus, this study joins the SPL engineering field with the proposal of a framework used for comparing different construction method, for identifying their drawbacks, and suggesting some ideas to solve them.

This paper is organized as follows. Our comparison framework is described in the next section and it is applied on four selected SPL's construction methods in the third section. In the fourth section we present our experimentation. The section 5 presents some related work. The section 6 concludes this work with our contribution and research perspectives.

2 The Framework

We have elaborated a framework to compare different approaches for the construction of SPL. The idea to consider a central concept (here the SPL) on four different points of view is largely inspired from [11], a work dealing with the Web Engineering. Defining a comparison framework has proved its effectiveness in improving the understanding of various engineering disciplines. Therefore, it can be helpful for the better understanding of the field of engineering SPLs. To elaborate our comparison framework, we have proceeded to an analysis of issues that are crucial for the amelioration of the SPL development method. As a result, our framework contains 12 attributes organized into 4 views (cf. Fig. 1) developed in the following subsections.

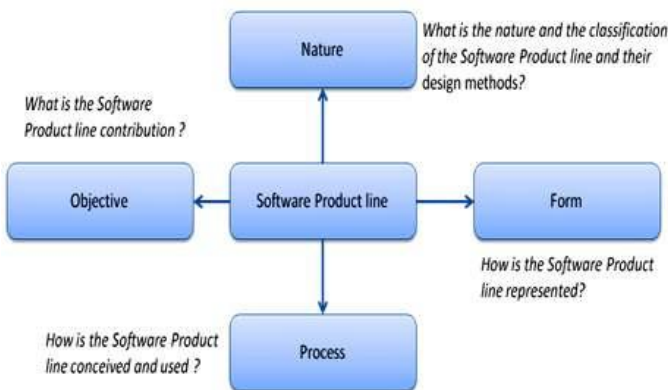


Fig. 1. Software Product Line comparison framework. This shows a figure consisting of our comparison framework with its different views.

Each view allows us to capture a particular aspect of SPLs engineering. Each view is characterized by a set of facets that facilitate understanding and classification of different aspects of SPLs engineering. The facets are defined using attributes which is described by a set of value for measuring and positioning the observed aspect. The facet approach was used in [10] to understand and classify approaches based on scenarios in the field of requirements engineering.

The multi-faceted and multi-view approaches adopted here, allow us to have an overview of the SPLs engineering. The views can show the variety and diversity of the different aspects of SPLs engineering. The facets provide a detailed description of these aspects.

The four views up of the comparison framework respond to four questions about the SPLs engineering:

- What are a method and an application?
- What is the objective assigned to the methods?
- How are represented the construction methods of SPLs?
- How to develop applications?

2.1 Objective View

This view captures why we should use a specific construction methods for SPL and what are the benefits retrieved from its practical application.

This view is related to the objectives we seek to achieve in the field of SPL engineering. A development approach of SPL can be classified in relation to its role. Such approaches were designed for different purposes and try to describe the process in different attitudes: descriptive, prescriptive or both of them. **Prescriptive methods** allow the prescription of how the process should or could be done. **Descriptive methods** allow the study of existing processes and provide explanations about how the development process was conducted. However, certain approaches may include the two strategies as cited in [18]: “Persons dealing with software processes can adopt two different attitudes: descriptive and prescriptive”. This aspect is captured by the **Goal facet**.

Moreover, since the applications change and evolve over the time, it is essential that the methods support these evolutions. The environment of SPL is constantly undergoing to technological, functional (in user needs), structural and behavioral change. Therefore, SPLs are constantly evolving which is important to consider in their development. The evolving systems are scalable systems that can be adapted and extended. This means that a design method for SPL must support evolution. This development should be managed in a controlled flexible and consistent manner. As with any software development, reuse is an important aspect, which can occur at any stage of development and at any level of abstraction and may involve any element of design and/or implementation. There is a growing interest to the identification of reusable design components and design patterns that capture the expertise and experience of designers. Now, the reuse is a part of the policy management methods in the organization. This aspect is captured by the **Policy Management methods facet**.

2.2 Form View

The form view deals with different aspects that describe the SPL. This view concerns the methods representation. Indeed, we focus in this view on some points of interest which are:

- What has to be represented?
- How will it be represented?
- At what level of abstraction?

In this view, we found many facets: Models, Notation and Abstraction level.

The **Models facet** describes the various models to describe the methods. This facet describes the various models proposed by the method i.e. the different aspects taken into account when designing a SPL. Most of existing design approaches for SPL considers the design phase as an activity that focuses on producing models. The **requirements modeling** for the product line is necessary to try to reach a flexible solution tailored to different user requirements for products of this line to build. The **architecture modeling** of SPL is indispensable for the construction of the product line. The architecture of product lines is an architecture consisting of components, connectors and additional constraints [17]. The role of the architecture of product lines is to describe commonality and variability of the products in the product line and to provide a general common structure. The derived products are the instantiations of the product line architecture and the architecture component. The **user model** is built in order to determine the characteristics of users. It represents the knowledge, objectives and / or individual characteristics such as preferences, interests and needs of users. The **product model** presents the specific features and functionality on which the product must respond.

The **Notation facet** concerns how these methods are represented. Indeed, this facet captures the notation nature used in the proposed method (standard, owner or mixed). Some of the studied works are based entirely on standards like UML, MDA...

The **abstraction facet** deals with the abstraction level where these methods are applied. Depending on this level, a SPL model is used as is or will be instantiated. At the level of abstraction of meta-types are all approaches that rely on mechanisms of meta-modeling.

2.3 Nature View

This view answers the « What » question. This means that we will develop facets concerning the internal structure and formalization of the SPL. In this view, we study the nature of SPL and their classification, as well as the definition of methods for their design. In this view, we focus on three facets: SPL's nature, derived products' nature and method's nature.

The first facet presents two types of SPL which are Integration-oriented SPL and open compositional [16]. **Integration-oriented** SPL methods are based on a centrally maintained reusable asset base. This reusable asset forms the basis for any derived products. The reusable assets contain the SPL architecture with a set of components and variation points.

Open compositional SPL methods are based on a distributed software ownership, different goals of parts owners and sub-contracting of part of the software. In this approach, product developers select partially integrated components and combine it with their own components. However, there is no pre-integrated platform and product developers are free to select the components which suit their needs from the available components.

The **facet products' Nature** can describe the different products resulting from SPL (component, service, constraints on the product, product description, architecture...).

The proposed research in the field of SPLs engineering are designed to explore different ways to structure the development process, using models and tools tailored to the needs and specificities of the SPL. Some classifications have tried to show the diversity of these approaches in terms of their level of abstraction, granularity and focus. The **method' nature facet** deals with these classifications (user-driven approaches, model-oriented approaches, user-centered approaches). The **user-driven approaches** try to identify and define the needs of target users through the step of requirements analysis. Two main techniques are used to determine the user needs: the use cases and the scenario-based approach. **Model-oriented approaches** are model-driven methods. Indeed, modeling the application domain is the starting point of their approach in the construction of SPL. In **user-centered approaches**, the user concept is privileged since the first step in the development process in order to capture the knowledge about the target audience of the future system and to model subsequently the scope of information pertaining to these users. Indeed, a step of user modeling is included in the life cycle approach to identify and describe the different classes of target users to determine the requirements for each category.

2.4 Process View

The process view considers different ways of construction methods for SPL conception and usage. Managing variability in SPLs occurs at different levels of abstraction during the development cycle of the product line. All possible variants decrease throughout the development phases. The more we advance in the development, fewer decisions are to be taken towards the possible variation.

A variation belonging to a particular level of abstraction, may give rise to one or more variations located in the lower levels. It must have traceability links among different levels of variation. This helps the identification of variation points to treat after selecting an option belonging to a particular level of abstraction. The choices made at different levels of abstraction can keep the most relevant variants. The number of variants depends from the nature of the system to build. For example, an ERP must be highly variable to suit the needs of several types of businesses.

The **Lifecycle Coverage facet** deals with the development cycle of a SPL. There is still no consensus on a general model of this development cycle. SPL engineering is defined in the literature [12] by distinguishing two levels of engineering: Domain Engineering and Application Engineering as presented in Fig. 2. **Domain Engineering** (The engineering for Reuse) corresponds to the study of the area of product line, identifying commonalities and variabilities among products, the establishment of a generic software architecture and the implementation of this architecture. Indeed, the domain engineering consists on the development and construction of reusable components known as asset (specification document, model, code ...) which will be reused for the products building. For this reason, the domain engineering is considered as development for reuse. **Application Engineering** (The Engineering by Reuse) is used to find the optimal use for the development of a new product from a product line by reducing costs and development time and improve the quality. At this level, the results of the domain engineering (characteristics model, generic architecture, and components) are used for the derivation of a particular product. This derivation corresponds to the decision-making towards the variation points. It is a development by reuse. The derivation of a particular product requires decisions or choices associated with different variation points detected.

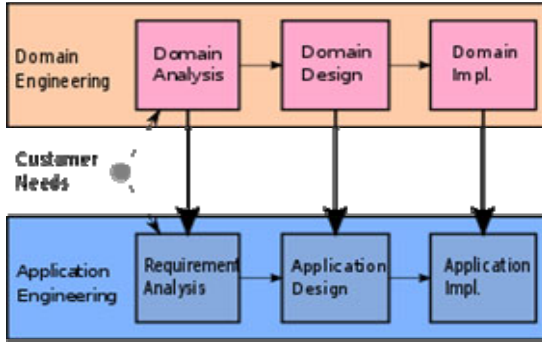


Fig. 2. Software Product line Process. This shows the development lifecycle of SPL.

The **construction technique facet** tries to captures the nature of the used techniques which are based on the instantiation of meta-models, assembly (components, services or COTS), languages and ad-hoc. **Instantiation of a meta-model** is based on the meta-modeling which consists on the identification of common and generic features of products and then represent them by a system of generic concepts. Such representation has the potential to 'generate' all applications sharing the same characteristics. **Assembly technique** consists on the building of a base of reusable components and the assembling of them in products derivation. The reusable components can be either code components or design component. **Language**: The software engineering community has used different languages to develop software applications. This technique has been adopted in the field of SPLs (i.e. compilation directives, template, inheritance...). A sample for the use of compilation directives is shown in Fig. 3. **Ad-Hoc**: Some construction approaches for SPLs are based on expression of the developers' experience. While this experience is not formalized and does not constitute a knowledge base available for the individual developers, we can say that such applications are the result of ad-hoc construction technique.

```

#ifdef VariantA          /* optional part */
#include CodeForVariantA
#endif
#ifdef VariantB1        /* alternative part */
#include CodeForVariantB1
#else
#ifdef VariantB2
#include CodeForVariantB2
#else
/* default functionality for VariantB goes here */
#endif
#endif
#endif
    
```

Fig. 3. The use of compilation directive. This shows an example of the use of some compilation directive to derive a variant of products.

The **Runtime support facet** permits the determination if the approach is supported by a tool. The world of development concerns, in addition to the construction of product line and derived products, necessary assistance for their implementation and

execution. Tool environment is necessary to help the implementation of methodological process and is also a part of development world problems.

The **adaptation facet** tries to find the different dimensions that can be adapted in a SPL: features, structures, behaviors and operating resources. Indeed, the heterogeneity of users is observed on several levels: their goals, their knowledge, skills, preferences in terms of features, product structures, behaviors or physical resources from which they access the application. Therefore, it is necessary that the product lines are adapted to their users.

3 Framework Application

Several methods for the construction of SPL have emerged in the literature. Before applying our comparison framework to these methods, we give their brief overview in the following sub-section.

3.1 Overview of Existing Methods

3.1.1 Van Gorp Method

Reference [3] considers the variability as the key to software reuse. The reusable software is distinguished from normal software by supporting various types of variability. Reference [3] has provided a terminology for reasoning about variation points. He described, too, the influence of SPLs approach in the development process. He presented, also, a method to identify, plan and manage the variability in a SPL. This method is based on identification of variation points through the creation of features diagrams. Then, an evaluation phase of variation point's properties is established. The binding of variation points to a variant is the last step of the method which consists in the instantiation of a class and assigns an instance to a property.

According to our framework, this method has as objective a prescriptive attitude to describe process and has the ability to manage the evolution and the reuse by the evaluation phase of variation points. It offers the product model as output and it uses proprietary notation (features diagrams). This user-driven method is based on oriented-integration strategy for the construction of the SPL. In its process, it uses the instantiation, language and Ad-Hoc construction techniques without a tool support.

3.1.2 Ziadi Method

Reference [5] proposes to model SPL variability in static diagrams (use case diagrams and class diagrams) and dynamic diagrams (sequence diagrams using the composition operators of UML 2.0). To ensure the consistency of the SPL model, there are structural rules expressed in OCL, which any SPL must respect. These constraints can be generic or specific to a SPL. Reference [5] proposes an approach for the products derivation (moving from the product line to a particular product) based on the automatic generation of state machines from sequence diagrams. This generation is possible with the use of algebraic specification.

This method supports the evolution and the reuse and offers the ability to construct architecture and product model using a standards notations (UML, OCL). As output, it makes product and architecture description and constraints on the products. This method is a model-oriented method and covers the entire product line lifecycle by using instantiation of model and Ad-Hoc techniques. It allows the adaptation of features, structures and behaviors.

3.1.3 Deelstra Method

Reference [4] proposes the use of model driven architecture (MDA) to manage variability in SPL as presented in Fig. 4. In this proposal, MDA is used as an approach to derive products in a particular type of SPL (configurable product line). The contribution of this study was to combine MDA with a configurable SPL and present the benefits of this relationship. In agreement with the management of variability, two main benefits of applying MDA to the product line engineering are identified, namely the independent evolution of domain concepts, the components of the product line, technology processing and infrastructure used and the use and management of variability as a solution to the problem of round transformations in MDA.

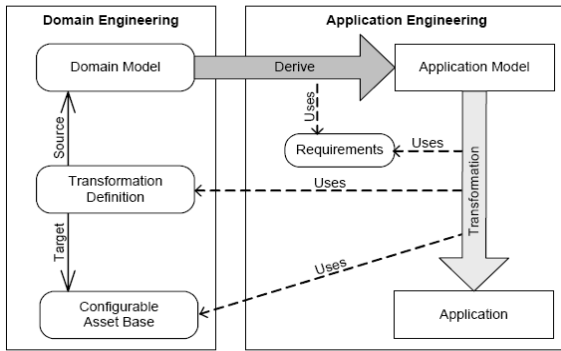


Fig. 4. The use of MDA in Product line engineering. This shows the integration of MDA in SPL lifecycle.

This method supports the evolution and reuse by the use of MDA. It proposes a prescriptive attitude to describe the software process. According to the form view, it produces the architecture and product model with a standard notation (MDA) and in meta-types as level of abstraction. The nature of the derived product of this model-oriented method is component, service, product description and architecture. Its process covers the entire lifecycle and uses instantiation as construction technique without a tool support. Finally, it tries to adapt the operating conditions by the use of MDA which can handle both the platform and technical variability.

3.1.4 Djebbi Method

The RED-PL approach [6] is developed to provide a response to the manner to ensure the satisfaction of the real user’s needs and the derivation of the optimal product requirements set (the right product to build). Reference [6] tries, through this approach, to take into account new stakeholders’ needs during requirements derivation for product lines. The RED-PL approach stands for “Requirements Elicitation & Derivation for Product Lines”. The matching process tries to do compromise between the product line requirements and the satisfaction of users’ needs. The matching process tries to do compromise between the product line requirements and the satisfaction of users’ needs.

This method aims for a both prescriptive and descriptive attitude with reference to the process description and it supports the evolution and reuse by the matching

process. According to the form view of our framework, it proposes a requirement model besides the product model in a proprietary notation (features diagrams) and a type abstraction level. This method is user-centred because the user concept is privileged. It tries to construct an oriented-integration SPL based on a centrally maintained reusable asset base. It covers only the domain engineering with a Ad-Hoc construction technique and tries to adapt the features and behaviors in agreement with user's needs.

3.2 Comparative Analysis within Framework

The various methods that we have presented approach the design and construction of SPL from different angles. We're going in the first sub-section to summarize characteristics of different methods according to the four views of our framework to pass in the second sub-section to present a number of shortcomings of these methods.

3.2.1 Evaluation Summary

The table in [21] presents a comparative analysis of the four selected SPL construction methods. Every method of those presented in sub-section 3.1 covers one particular aspect of the construction of SPL. Reference [3] tries to manage the variability variation point in product line by using features diagrams. Reference [5] is focusing the design aspect by using UML profile (to model SPL and then manage variability in static and dynamic UML diagrams) and OCL constraints (to specify generic or specific constraints on products). Reference [4] proposes the use of MDA to handle both the platform and technical variability by the focus on the design aspect. Reference [6] tries to cover the requirement aspect by the requirement elicitation and the study of constraints in order to ensure the satisfaction of the real user's needs and the derivation of the optimal product.

3.2.2 Drawbacks of Existing Method

The framework analysis allows identifying the following main drawbacks of existing SPL construction methods. We realize that we have a lack of sufficient tool support for them and for their interactivity with their users. The SPL approaches themselves are not enough automated for deriving automatically a product from a SPL. Moreover, some of the proposed methods are using proprietary notations which can handle some problems of standardization and interoperability... In addition, these methods didn't cover all aspects of SPL engineering. Indeed, every method tries to focus on a particular part of SPL construction process. For example, [5] focus his works on the design of SPL and the derivation of products. Reference [6] is working on requirements engineering for SPL to take into consideration the real users needs. In other hand, all these methods offer a prescriptive process that dictates to the designer to perform activities to achieve a particular task. This guides the designer. However, it restricts its participation in the design process. Indeed, such process does not afford him the opportunity to be active in choosing alternatives for achieving its objectives. Finally, in these method, apart [6] ones, the problem is the matching between users' needs and the product offered by developers. The difficulty in mapping lies in the differing languages in which the two parties involved, developers and customers are accustomed to express their self. Many writers have observed that there is a "conceptual mismatch" [13] [14] [15]. Indeed, the developer is placed in operational level, while on the other side customers are placed at the intentional level.

4 Experimentation

In this section we briefly present our actual axis of research which is the experimentation. Actually we try to analyze many existing open source ERP in order to find the best solution with possibilities of evolution. We found two major solutions which are Compiere and Openbravo. Every one of these solutions has many possibilities of customizing and extensions by the adding of new components (reports, helpdesk, task management, payroll...) or customizing the existing ones (Customer Relations Management, Partner Relations Management, Supply Chain Management, Performance Analysis, Web Store...) as presented in Fig. 5 and Fig. 6. Since their inception, Compiere and Openbravo have provided an alternative to expensive and closed ERP systems from Oracle, SAP...

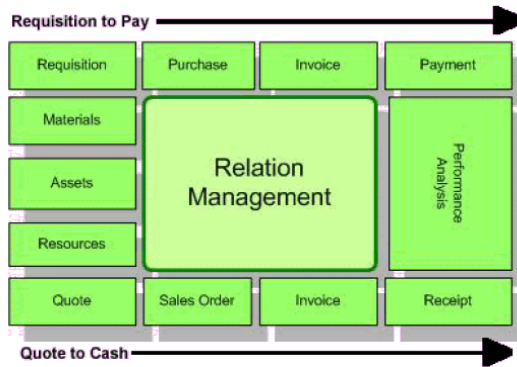


Fig. 5. Domain covered by ERP. This shows management possibilities in an ERP solution.

Compiere and Openbravo are two complete business solutions for small-to-medium enterprises, particularly those in the service and distribution industry, both retail and wholesale. These solutions have an integrated Web Store, covering material management, purchasing sales, accounting, and customer relations management.

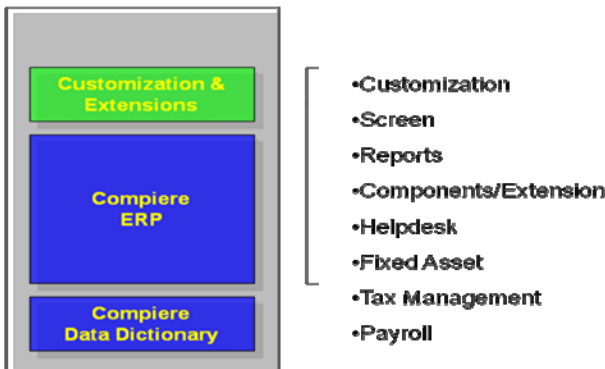


Fig. 6. Compiere specificities. This shows possibilities of customization and extensions in the Compiere ERP solution.

Our methodology consists, as first step, on the evaluation of Compiere and Openbravo. Also, we try to analyze the functional and technical choices done in these solutions to detect the possibilities of evolution according to the real requirements of users by the testing and getting them into production. And finally, we aim to customize and extend the ERP to meet these specific requirements. This final step needs an intentional study to avoid "conceptual mismatch" [13] [14] [15] between developers and customers. Indeed, the developer is placed in operational level, while on the other side customers are placed at the intentional level.

5 Related Work

In this section we briefly present some of the research literature related to comparison framework and Software Product Line. The construction of comparison framework for a specific domain is not new. This technique has proven its effectiveness in improving the understanding of various engineering disciplines, such as requirements engineering [19], method engineering [20], and web engineering [11]. The facet approach centerpiece of our comparison framework was used in [10] to understand and classify approaches based on scenarios in the field of requirements engineering.

Otherwise, the domain of software product line is a rich domain with many proposals and techniques in the literature [3] [4] [5] [6] [7] [8] [9]... In these works, we found the use of many techniques for implementing variability and for the construction of software product line like the use of features diagrams, UML diagrams, MDA, ontology, programming techniques... In this domain, many paths are possible to try to implement variability and to construct with success an SPL.

6 Conclusion and Future Work

In this paper, our contribution was the definition of a comparison framework which has allowed identifying the characteristics and drawbacks of some existing methods for the construction of Software Product Line.

The suggested framework allows a comparison structured in four views. It was build to respond to the following purposes: to have an overview of existing Software Product Line construction methods, to identify their drawbacks and to analyze the possibility to propose a better method.

Based on this framework analysis, we propose to improve the method used for software product line construction in order to overcome the following drawbacks of existing methods by proposing a tool support to improve interactivity with users. Also, we will try to cover the overall lifecycle of software product line. To avoid the conceptual mismatch, we will try to establish the matching between users' needs and the product offered by developers by the expression of users' needs in an intentional way.

Our future works include a case study of a particular software product line ERP to try to identify variability in this solution. After that, we will try to change in abstraction by studying the ERP in an intentional level to attempt to find the real users' needs and to model the user.

References

1. SEI Product Line Hall of Fame web page, http://www.sei.cmu.edu/productlines/plp_hof.html
2. Weiss, D.M., Lai, C.T.R.: Software Product-Line Engineering. In: A Family-Based Software Development Process, Addison-Wesley, Reading (1999)
3. Van Grup, J.: Variability in Software Systems, the key to software reuse. University of Groningem, Sweden (2000)
4. Deelstra, S., Sinnema, M., van Gurp, J., Bosch, J.: Model Driven Architecture as Approach to Manage Variability in Software Product Families. In: Proceedings of the Workshop on Model Driven Architecture: Foundations and Applications, University of Twente (2003)
5. Ziadi, T.: Manipulation de Lignes de Produits en UML. Université de Rennes, Rennes (2004)
6. Djebbi, O., Salinesi, C.: Single Product Requirements Derivation in Product Lines. In: CAiSE (2007)
7. Haugen, Ø., Møller-Pedersen, B., Oldevik, J., Solberg, A.: An MDA-based framework for model-driven product derivation. In: Software Engineering and Applications, MIT, Cambridge (2004)
8. Sinnema, M., Deelstra, S., Nijhuis, J., Bosch, J.: COVAMOF: A Framework for Modeling Variability in Software Product Families. In: Nord, R.L. (ed.) SPLC 2004. LNCS, vol. 3154, pp. 197–213. Springer, Heidelberg (2004)
9. Lee, J., Kang, K.C.: A Feature-Oriented Approach to Developing Dynamically Reconfigurable Products in Product Line Engineering. In: Software Product Line Conference (2006)
10. Rolland, C.: A Comprehensive View of Process Engineering. In: Pernici, B., Thanos, C. (eds.) CAiSE 1998. LNCS, vol. 1413, pp. 1–24. Springer, Heidelberg (1998)
11. Selmi, S.: Proposition d'une approche situationnelle de développement d'applications Web. Université de La Manouba : Ecole Nationale des Sciences de l'Informatique (2006)
12. Czarnecki, K., Eisenecker, W.: Generative Programming: Methods, Tools, and Applications. Addison-Wesley, Reading (2000)
13. Woodfield, S.N.: The Impedance Mismatch Between Conceptual Models and Implementation Environments. In: ER 1997 Workshop on Behavioral Models and Design Transformations: Issues and Opportunities in Conceptual Modeling, UCLA, Los Angeles (1997)
14. Object Matter, <http://www.objectmatter.com>
15. Kaabi, R.: Une Approche Méthodologique pour la Modélisation Intentionnelle des Services et leur Opérationnalisation. Université de Paris I, Sorbonne (2007)
16. Van Grup, J., Prehofer, C., Bosch, J.: Comparing Practices for Reuse in Integration-oriented Software Product Lines and Large Open Source Software Projects. Software: Practice & Experience, Wiley (2009)
17. Bass, L., Clements, P., Kazman, R.: Software Architecture in Practices. Addison-Wesley, Reading (1998)
18. Lonchamp, J.: A structured Conceptual and Terminological Framework for Software Process Engineering. In: Proceedings of IEEE International Conference on Software Process (1993)

19. Jarke, M., Pohl, K.: Requirements Engineering: An Integrated View of Representation, Process and Domain. In: Proceedings 4th Euro. Software Conf., Springer, Heidelberg (1993)
20. Deneckère, R., Iacovelli, A., Kornysheva, E., Souveyet, C.: From Method Fragments to Method Services Exploring Modeling Methods for Systems Analysis and Design (EMMSAD 2008). Montpellier, France (2008)
21. Ouali, S., Kraiem, N., Benghezala, H.: A Comprehensive View of Software Product Line. In: ICMSC, Egypt (2010)

Collecting the Inter Component Interactions in Multithreaded Environment

Arun Mishra, Alok Chaurasia, Pratik Bhadkoliya, and Arun Misra

Computer Science & Engineering, MNNIT,
Allahabad (U.P.), India

{rcs0802,cs074037,cs074003,akm}@mnnit.ac.in

Abstract. Whenever a new upgrade in software is made or a new component is loaded, that changes may impact the existing system's execution. Our objective is to define an approach that can be applied to on-line validation of component integration in autonomous system (AS). One important means of assuring the validation of component interactions is through analyze interactions among different components. Several tools have been developed to capture interactions among components. These tools do not support all the requirements for building interaction diagram in multithreaded environment. We have developed a technique to capture the run-time components interactions using .NET CLR mechanism. By this technique, we have been able to successfully capture the interactions among components across all application threads. A case study has been carried out on multithreaded self-adaptive system.

Keywords: Component Interactions, Runtime, Self-adaptive System, Multithreaded System, Trace.

1 Introduction

Component based system has been widely used in various application domains. However, lack of information about components developed by other developers and the complex interactions among components lead to validation challenges. As a matter of fact, consider the situation that 'CompB' calls 'Func2' of the 'CompC', while 'CompA' has already invoked 'Func1' of the 'CompC'. Similarly, 'Func3' of the 'CompD' gets a call from 'CompC'. In this situation it is difficult to recognize the thread that gave rise to the invocation of 'Func3' towards 'CompD'.

Multithreaded system cannot be depicted using standard UML representation as shown in Fig 1. This introduced the need for identifying the thread originator of the each event; our objective is to define an approach to visualize the inter component interactions in such component based multithreaded systems.

A general profiler works with CLR (middleware of .NET Framework) and collects all the traces of execution at the runtime. This type of profiling reduces the performance of the system by including unnecessary information (such as predefined .NET classes' interactions, intra-component interactions etc.). We have developed a technique to trace the component interactions at runtime, across all threads of the system with minimal overhead. The interactions thus traced are used to build 'Trace Diagram' which inherits UML sequence diagram standards and extends some rules.

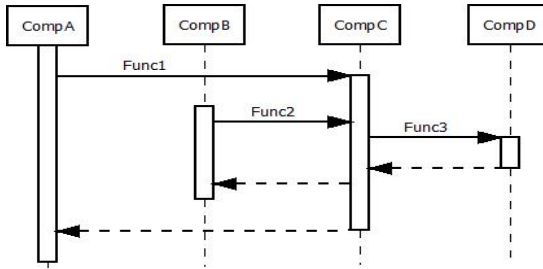


Fig. 1. Concurrent Method Invocation Problem

Our approach can be applied to runtime analysis of component integration in self-adaptive system. In the self-adaptive system, components are dynamically replaced by a similar component in terms of functionality, whenever the current working component leads to reduction of system's performance. A case study has been carried out on a component based multithreaded self-adaptive system in which the application uploads file to a server over Internet under its normal behavior. When bandwidth is found below a certain threshold value, the application adapts a new component called ZIPP that compresses the file and uploads compressed version of the file to the server. Multiple threads are running concurrently in the system for different tasks that includes observing and analyzing the bandwidth available to the application, uploading the file to server and for interaction with the user. The custom profiler and distiller are implemented in Visual C++ using .NET Profiling API. Traces are collected at runtime and stored into a trace file at the end of the execution and can be inspected. A trace diagram has been given in the end, highlighting the inter component interactions in multithreaded environment of the case study file transfer application.

The paper is organized in the following manner: section 2 presents the literature review in field of self-adaptive system and validation; section 3 presents our approach towards the objective; section 4 discusses the runtime profiling in terms of customized profiler and distillation; section 5 presents case study, in which we show the trace diagram; section 6 finds place for conclusion and future work.

2 Literature Review

Automated software (a kind of complex software) has been successfully applied to a diversity of tasks, ranging from strong image interpretation to automated controller synthesis [1] [2]. Due to complexity of analyzing the complex system (because of multithreaded or concurrent nature and the black box nature of components) several different approaches to monitor their behavior have been proposed. An approach used in [3], authors instrument the architectural description, and not the middleware and they require the developers to define a set of rules used to analyze the traces. Atanas R. et al. in their work introduced the UML based technique to generate the different aspect of component interactions after the modification [4] and used those results to define testing goals. Felipe C. et al. worked on reverse engineering environment to support extraction and detection of implied scenarios from dynamic system information captured during the execution of existing concurrent applications [5]. The main contribution of their work is a practical demonstration of applying filters to the set of

captured events. Atanas R. et al. generalize the technique for dynamic analysis of multi-threaded programs by keeping multiple traversal stacks [6]. Giovanni M. et al. gave the idea to register the invocation and return timestamps of methods calls between the components [7]. Their tool relies on dynamically instrument methods during class loading.

3 The Approach

Multithreaded system includes several threads which are executed concurrently. The problem comes in visualizing the exact behavior of the system under various circumstances. Our approach lies in tracing the runtime components interactions on the basis of runtime customized profiling, distillation. Custom profiler works along with distiller and is responsible for the collection of traces at runtime for the component based multithreaded system. These traces are interpreted using trace diagram. Multithreading increases the degree of complexity in the system.

To demonstrate the concurrent method invocations in case of multithreading system we extend the UML standard in order to capture an execution trace that can be represented using a trace diagram as shown in Fig 2. This solution successfully overcomes the problem present in Fig 1.

Following Rules represents the extended rules to explain the process in multithreaded environment.

- Different colors are used to represent each thread, Object activation lanes and method calls and returns.
- The period of time in which we do not know what the thread is doing is represented with a vertical waved line from the time of return arrow starting with temporal gap to the time of the first known method call.
- Thread start and end are marked with a filled circle.

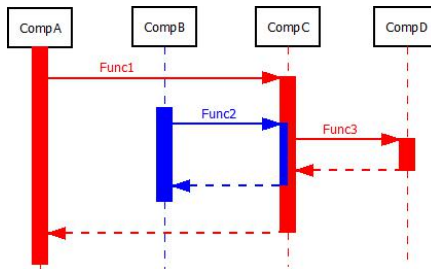


Fig. 2. Concurrent Method Invocation Solution

4 Runtime Profiling

Runtime profiling involves the technique to construct the exact execution flow of each thread. For the profiling purpose we plan to trace information about the behavior of the system during its execution. To profile the system at the runtime we need profiling activity at the middleware level [8].

In proposed work, profiler directly works with CLR (middleware of .NET). CLR provides logical place to insert hooks to see exactly what the system up to [9]. Depth of the result obtained after profiling is too high; so, we designed a customized profiler that includes distillation process.

4.1 Customized Profiler

We designed a customized profiler to profile the system, when its components are loaded at run time. By exploiting a hook provided by the CLR, it is possible to register our customized profiler that impacts each interaction and reports only the events of our interest in multithreaded environment. This approach provides the user with high level of transparency. This approach has very low overhead between the events and its tracing. In order to reduce the impact of information overload, customized profiler has the distillation on the events.

4.2 Distillation

The reason for us to use distillation process for the profiling is that our interests are confined only to inter-component interactions and not in intra-component interactions. Boundary calls allows us to distill lot of intra-component interactions. A boundary method call is a call whose receiver if lies in component A, then its caller cannot be component A. Boundary calls facilitates us to reduce the state space of method call combinations. We apply three types of distillers.

- Methods call belong to utility classes
- Calls to class constructor
- Calls to inner methods.

These distiller types, when applied to the execution traces gives the different levels of abstraction. At the first level we intend to better confine the system by preserving only those method calls that are directly involved in its implementation. At this level profiler distills out all method calls belonging to utility classes i.e. classes that offer common services to several other classes of the target system. The second and third levels distilled out further method calls considered irrelevant from the perspective of

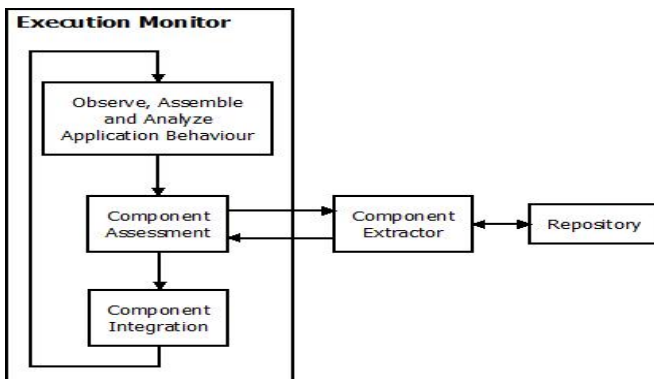


Fig. 3. Framework for Component Based Self Adaptive System

components interactions, namely, invocation of class constructors and invocations of inner methods, respectively. Distillation is intended to better confine the target system under analysis. These distiller types can vary as per user's requirements and need.

5 Case Study: Applying the Approach to the Component Based Multithreaded Self-Adaptive System

In self-adaptive systems where condition for the execution is changing continuously, component performance may degrade dramatically and may cause reduction of system performance. In our previous work [10], we have introduced a framework that automates the component extraction from repository and integration process. We have presented a basic tree equivalency algorithm to compare components.

Fig 3 represents a framework for component based self-adaptation at run-time [10]. Framework is represented as a collection of integrated components, allowing run-time adaptability.

The 'Execution Monitor' continuously monitors the behavior of the application while it is running. The major functions of the system are:

- Observe, assemble and analyze application behavior concurrently with application execution in a different thread.
- If diversion is found with respect to expected behavior of application, the adaptation strategies must reason about the impact of diversion and extract an appropriate component from the 'Repository', with the help of 'Component Extractor' that suits the current context.
- Integration of the new component that includes loading the new one, unloading the old (faulty) one, and restarting services to assure the functionality of the application.

5.1 Test Application

File transfer over internet is a day to day job for most computer users. The network bandwidth available to a computer node in a private network connected to internet, changes significantly due to request overhead by other computer nodes in the private network. When bandwidth available is low, file transfer takes longer time than usual. One approach to overcome this problem is to transfer compressed version of the file. Since, compression of the file reduces the size of file.

We have implemented a multithreaded self-adaptive file transfer client system, which uploads files on server over internet. The client system uploads a file on the server, but when bandwidth is found below a known threshold value, a component is dynamically loaded from the 'Repository' which compresses the file and uploads the compressed file on the server. The system can be categorized in three segments. These are Execution Monitor, Extractor and Repository.

Execution Monitor. The segment consists of mainly four components which are 'Client', 'Monitor', 'NormalUploader' and 'BandwidthChecker'. The Client is a

Graphical User Interface (GUI). The Monitor automates the self-adaptive system. Whenever a failure occurs in the system, the Monitor interacts with the Extractor and integrates a new component from the Repository, in order to assure normal behavior. The NormalUploader and BandwidthChecker components execute concurrently in different threads. The NormalUploader uploads a file to server over internet and the BandwidthChecker assembles, evaluates and examines bandwidth available to the client computer. If bandwidth is encountered below a known threshold value, BandwidthChecker stops normal file transfer and generates system failure event.

Extractor. The segment consists of only one component i.e. Extractor. It searches components in the Repository, on system failure. To fetch best suitable component available from the Repository, we use the Abstract Syntax Tree (AST) based approach as explained in previous work [10].

Repository. The segment consists of large number of components. For example it contains a ZIPP component. The system will adapt ZIPP component when NormalUploader component has been stopped due to bandwidth below known threshold value. The ZIPP component compressed the normal file and uploads the compressed file on the server. So, when client computer suffer from bandwidth below threshold, the Monitor integrates the ZIPP component from Repository with the help of the Extractor component.

5.2 The Custom CLR Profiler

One means by which we can observe the .NET runtime in action is by using the profiling API provided by .NET Framework [11]. The profiling API consists of COM interfaces such as callback interfaces (ICorProfilerCallback [12] & ICorProfilerCallback2 [13]) and info interfaces (ICorProfilerInfo [14] & ICorProfilerInfo2 [15]). To profile .NET 2.0 or later applications, implementation of callback interfaces are required; the info interfaces can be used to get information while inside event callbacks.

We have designed 'CProfiler' class which implements 'ICorProfilerCallback2' interface and maintains 'CComQIPtr' smart pointers [16] to info interfaces. The following code snippet represents code for the same in VC++ language.

```
class CProfiler : public ICorProfilerCallback2
{
    CComQIPtr<ICorProfilerInfo> pICPInfo;
    CComQIPtr<ICorProfilerInfo2> pICPInfo2;
...}
```

Initialize and Shutdown events. The CLR raised 'Initialize' and 'Shutdown' events of ICorProfilerCallback when test application is started to execute and terminate, respectively. In 'Initialize' event handler, registration of events (such as Assembly load/unload, Thread created/destroyed) are done by calling SetEventMask method;

and hooks are set for Function Enter/Leave by calling SetEnterLeaveFunctionHooks2 methods.

The ‘Shutdown’ event handler will generate trace file. The trace file is useful to generate trace diagram.

AssemblyLoadFinished event. CLR raises the ‘AssemblyLoadFinished’ event when an assembly load is finished by CLR and passes its AssemblyID as parameters. GetAssemblyInfo method can be used to get its name. The following code snippet (in Visual C++) depicts the filtration of Utility Assemblies.

```
pICPInfo->GetAssemblyInfo(aId, BUFSIZE, &dummy, name, &aId, &mId);
aMap[aId] = FilteredAssembly(W2A(name));
```

In above code snippet ‘FilteredAssembly’ returns true if assembly is not system utility, otherwise false.

Function Enter/Leave hooks and Filtration. CLR raises events on function enter and leave, which is handled by ‘FunctionEnter’ and ‘FunctionLeave’ hooks. The ‘GetCurrentThreadID’ method provides the ‘ThreadID’ of current thread. The ‘GetTokenAndMetaDataFromFunction’ method gives required information of function. So, using assembly and class name profiler will filter out utility class methods and further intra-component interactions;

Thread Created/Destroyed. CLR raises ‘ThreadCreated’ and ‘ThreadDestroyed’ events when a thread is created and destroyed, respectively; which passes the ‘ThreadID’ as argument. The profiler maintains separate method stack for each thread. So, the new stack allocated in ‘ThreadCreated’.

5.3 Profiler Launcher

The CLR is the engine that drives .NET applications. When the CLR begins a process, it looks for two environment variables, COR_ENABLE_PROFILING and COR_PROFILER. If COR_ENABLE_PROFILING variable is set to 1, then CLR should use a profiler. Since profilers are implemented as COM objects, COR_PROFILER variable will be set to the GUID of the ‘CProfiler’ class. The following code snippet (in C#) depicts the Main method of Profiler Launcher application.

```
ProcessStartInfo psi = new ProcessStartInfo (exeFilePath);
psi.EnvironmentVariables.Add("COR_ENABLE_PROFILING", "1");
psi.EnvironmentVariables.Add("COR_PROFILER", PROFILER_GUID);
Process.Start(psi);
```

The next section shows the Trace File generated as a result of customized profiling.

5.4 Result: Trace File of Scenario

The trace file entries given below can be interpreted under following headers.

TimeStamp : Event ClassName :: MethodName : ThreadID

```

0: Initialize
3778285414: Enter Client :: Main : 5967392
12949253809: Enter Client :: browseButton_Click : 5967392
17998621752: Leave Client :: browseButton_Click : 5967392
20354749165: Enter Client :: uploadButton_Click : 5967392
20430626465: Enter Monitor :: startMonitor : 118454736
20614593492: Leave Client :: uploadButton_Click : 5967392
20717467072: Enter BandwidthChecker :: StartCheck : 130654872
21044261842: Enter NormalUploader :: Upload : 118454736
21693624120: Enter BandwidthChecker :: IsBelowThreshold : 118454736
21693729410: Leave BandwidthChecker :: IsBelowThreshold : 118454736
22497388195: Enter BandwidthChecker :: IsBelowThreshold : 118454736
22497510535: Leave BandwidthChecker :: IsBelowThreshold : 118454736
22497576697: Leave NormalUploader :: Upload : 118454736
22563788473: Enter Extractor :: Extract : 118454736
22736080821: Leave BandwidthChecker :: StartCheck : 130654872
24171053766: Enter Repository.Zipp :: Upload : 118454736
24171151952: Leave Repository.Zipp :: Upload : 118454736
24171232190: Leave Extractor :: Extract : 118454736
24171293043: Leave Monitor :: startMonitor : 118454736
24190385691: Enter Client :: Dispose : 5967392
24358378565: Leave Client :: Dispose : 5967392
24374318829: Leave Client :: Main : 5967392
24418540126: Shutdown
Thread :: 5967392
409937: Thread Created
Thread :: 118454736
20365516900: Thread Created
24242300496: Thread Destroyed
Thread :: 130654872
20716281709: Thread Created
22736347729: Thread Destroyed

```

5.5 Interpretation: Trace Diagram of Scenario

Fig 4 shows the trace diagram of scenario, where first the ‘User’ requests for uploading a file using ‘uploadButton_Click’ event to the ‘Client’. The ‘Client’ passes this request to the ‘Monitor’ using ‘startMonitor’ method call in another thread than the normal UI thread. Then, ‘Monitor’ invokes two methods concurrently in different threads, one for uploading the file and another thread to check if bandwidth below threshold. The diagram depicts that the normal file transfer fails before its completion because the bandwidth was found below the threshold value, so, the system adapts itself and switches to ‘Zipp’ after extracting it from ‘Repository’.

Note: The timestamp used in the trace diagram is relative to ‘Initialize’ event. Therefore, no unit of timestamp mentioned in the trace diagram.

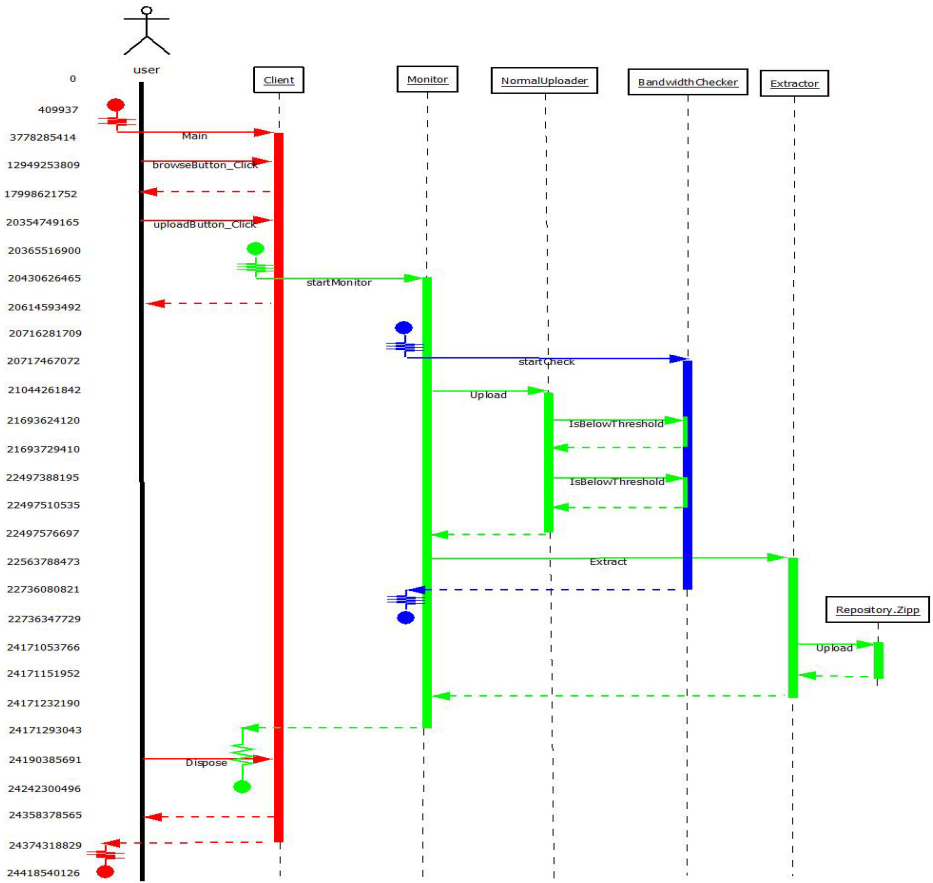


Fig . 4. Trace Diagram of Scenario

6 Conclusion and Future Work

We have presented and implemented an approach in that we have merged the strengths of system’s instrumentation with their execution, to specify the complex concurrent nature of software. Moreover, our work defined the approach that collects the execution flows of concurrent threads in trace file. Trace file have been used to casually describe the scenario of execution of software. In this paper we have proposed the approach that visualized the execution of concurrent threads. This can show appealing opportunity, particularly for the case of on-line validation of automated software.

By instrument the system at the middleware level, we could assess the parallelism of threads execution in complex software. Their feedback in terms of trace file is extremely useful in order to analyze complex nature of the software.

In future we will re-engineer the visualization software in order to increase the degree of usability and customization for the end-user. Further future work will include building the framework for automatic translation of trace file into formal modeling that may sound the possibility to include building the technique for the on-line verification of complex nature systems.

References

1. Garlan, D., et al.: Rainbow: Architecture-Based Self-Adaptation with Reusable Infrastructure. *Computer* 37(10), 46–54 (2004)
2. Oreizy, P., et al.: An Architecture-Based Approach to Self-Adaptive Software. *IEEE Intelligent Systems* 14(3), 54–62 (1999)
3. An Approach for Tracing and Understanding Asynchronous Systems. ISR Tech. Report UCI-ISR-02-7 (2002)
4. Rountev, A., Kagan, S., Sawin, J.: Coverage criteria for testing of object interactions in sequence diagrams. In: *Fundamental Approaches to Software Engineering* (2005)
5. Kramer, J., et al.: Detecting Implied Scenarios from Execution Traces. In: *WCRE*, pp. 50–59 (2007)
6. Rountev, A., Kagan, S., Gibas, M.: Static and dynamic analysis of call chains in Java. In: *International Symposium on Software Testing and Analysis*, pp. 1–11
7. Malnati, G., Cuva, C.M., Barberis, C.: JThreadSpy: teaching multithreading programming by analyzing execution traces. In: *International Symposium on Software Testing and Analysis*, pp. 3–13 (2007)
8. Bertolino, A., Muccini, H., Polini, A.: Architectural Verification of Black box Component Based Systems. In: *Proceedings of International Workshop on Rapid Integration of Software Engineering Techniques, Switzerland* (September 13-15, 2006)
9. Under the Hood: The .NET Profiling API and the DNProfiler Tool, <http://msdn.microsoft.com/en-us/magazine/cc301725.aspx> (accessed November 30, 2009)
10. Mishra, A., Mishra, A.K.: Component assessment and proactive model for support of dynamic integration in self-adaptive system. *ACM SIGSOFT Software Engineering* 39(4), 1–9 (2009)
11. Profiling (Unmanaged API Reference), <http://msdn.microsoft.com/en-us/library/ms404386.aspx> (accessed November 29, 2009)
12. ICorProfilerCallback Interface, <http://msdn.microsoft.com/en-us/library/ms230818.aspx> (accessed December 2, 2009)
13. ICorProfilerCallback2 Interface, <http://msdn.microsoft.com/en-us/library/ms230825.aspx> (accessed December 2, 2009)
14. ICorProfilerInfo Interface, <http://msdn.microsoft.com/en-us/library/ms233177.aspx> (accessed December 2, 2009)
15. ICorProfilerInfo2 Interface, <http://msdn.microsoft.com/en-us/library/ms231876.aspx> (accessed December 2, 2009)
16. CComQIPtr class, <http://msdn.microsoft.com/en-us/library/wc177dxVS.80.aspx> (accessed December 5, 2009)

On the Benefit of Quantification in AOP Systems-A Quantitative and a Qualitative Assessment

Kotrappa Sirbi¹ and Prakash Jayanth Kulkarni²

¹ Department of Computer Science & Engineering, K L E's College of Engineering & Technology, Belgaum, India
kotrappa06@gmail.com

² Department of Computer Science & Engineering, Walchand College of Engineering, Sangli, India
pjk_walchand@rediffmail.com

Abstract. In this paper, we support the statement that the most favorable uses of aspects happen when their code relies extensively on quantified statements, i.e., statements that may affect many parts of a system. When this happens, aspects better contribute to separation of concerns, since the otherwise duplicated and tangled code related to the implementation of a crosscutting concern is confined in a single block of code. In this paper we try to provide some insight both qualitative and quantitative arguments in favor of quantification. We describe an Eclipse plugin, called ConcernMetrics and AOPMetric tool that estimates the proposed metrics directly from the object-oriented code of an existing system, i.e., before crosscutting concerns are extracted to aspects. The strategy is applied to the JHOTDRAW open source project and study, how AOP system quantification provides help to developers and maintainers to decide in a cost-effective way.

Keywords: Aspect Oriented programming (AOP), Quantification, Separation of Concerns, Crosscutting Concerns, Metrics.

1 Introduction

Aspect-Oriented programming (AOP) [1] is an emerging paradigm that promises to improve software design and promotes reuse. Recently, Aspect-Oriented Software Development (AOSD) has shifted from an experimental effort to a more mature software engineering tool. However, the characterization of a good aspect is still not well defined. The existing research on object oriented quality provides a basis for the evaluation of aspect-oriented design. Several studies have shown that there is a high correlation between metrics and extern manifestations of quality such as the maintenance effort and the number of defects found pre- and post-release.

Aspect Oriented Programming (AOP), in addition to separating the different concerns during software development, can be seen as a way of overcoming many of the problems related to software evolution [1]. At the same time, AOP may benefit from tools and techniques that automate software evolution. Real-world software needs to evolve continually in order to cope with ever-changing software requirements. Many

Lehman identified this characteristic in his so-called first law of software evolution [1], which addresses continuing change: a program that is used must be continually adapted else it becomes progressively less satisfactory. This need for software to evolve continuously poses important challenges for software engineers. Advanced automated software engineering techniques and tools are needed to improve software evolution support. Two important techniques that will be investigated are software restructuring and aspect-oriented programming [1].

2 Background and AO Motivating Systems

2.1 Background

Aspect-oriented programming (AOP) targets a relevant problem in software engineering: the modularization of crosscutting concerns [2]. In order to tackle this problem, AOP extends traditional programming paradigms, such as procedural and object-oriented, with powerful abstractions. For example, AspectJ—the most mature aspect-oriented language—extends Java with new modularization abstractions, including join points, pointcuts, advices, and aspects [3]. In AspectJ, an aspect defines a set of pointcut descriptors that matches well-defined points in the program execution (called join points). Advices are anonymous methods implicitly invoked before, after, or around join points specified by pointcuts.

Despite the increasing number of qualitative and quantitative assessments of AOP [4], [5], [6], [7], [8],[9] there is still no consensus on the impact and the real benefits of using aspects—as proposed by AspectJ—to modularize crosscutting concerns [10]. In this paper, we contribute to such assessment efforts by emphasizing the benefits of a well-known characteristic of AOP: the notion of quantification [11]. This notion is used to designate statements that have effect at several parts of the code. We argue that one of the most favorable uses of aspects happen when their code relies extensively on quantified statements. When this happens, aspects better contribute to separation of concerns, since the otherwise duplicated and tangled code related to the implementation of a crosscutting concern is confined in a single block of code.

Zhang and Jacobson use a set of object-oriented metrics to quantify the program complexity reduction when applying AOP to middleware systems [11]. They show that refactoring a middleware system (23KLOC code base) into aspects reduces the complexity and leads to a code reduction of 2–3 %, which is in line with results available through research. Garcia et al. analyzed several aspect-oriented programs (4–7KLOC code base) and their object-oriented counterparts [12], [13]. They observe that the AOP variants have fewer lines of code than their object-oriented equivalents (12% code reduction). Zhao and Xu propose several metrics for aspect cohesion based on aspect dependency graphs [10]. Gelinas et al. discuss previous work on cohesion metrics and propose an approach based on dependencies between aspect members [14]. All of the above proposals and case studies take neither the structure of crosscutting concerns nor the difference between basic and advanced crosscutting mechanisms into account. Lopez-Herrejon et al. propose a set of code metrics for analyzing the crosscutting structure of aspect-based product lines [15]. They do not consider elementary crosscuts

but analyze crosscutting properties of entire subsystems (features), which may have a substantial size. Thus, the crosscutting structure of a feature can be homogeneous, heterogeneous, or any value in between the spectrum of both. They do not distinguish between basic and advanced dynamic crosscuts.

2.2 AO Motivating Systems

In order to make review of claims, we have selected the analysis of a diverse selection of AspectJ programs. The first 7 are small programs (<20KLOC); the last 4 are larger (_20KLOC) [16]. In table1 and figure1&2, the programs are listed from smallest (Tetris) to largest (Abacus). It also indicates in Table 1 if the program was developed from scratch (Type S), or if it was an AOP refactoring of an existing application (Type R). In certain situations, it will be consider only a subset of the eleven programs, e.g., large-sized programs only, in order to explore the specific properties of an individual program or subset of programs [16].

The aspect-oriented version of both systems has been independently developed by various researchers, in order to illustrate the application of aspect-oriented refactorings, i.e., step-by-step transformations that prescribe how to modularize crosscutting concerns. Furthermore, the aspects implemented in the mentioned systems make opposite uses of quantification.

First, we illustrate the use of quantified statements in the JHotDraw¹ migrated into AspectJ AJHotDraw². JHotDraw is a framework offering two dimensional drawing facilities. It consists of approximately 18,000 non-commented lines of code and 2,000 methods. JHotDraw contains a quite well structured source code, because it has been written as an example to show good practice in using design patterns. So, it is a good candidate as a case study for aspect mining. In fact, most of the concerns are expected to be already separated. Those still tangled are probably associated to intrinsic limitations of OOP and aspects may be a good alternative for them. Then, next, we assess the benefits of quantification in such systems, using a suite of separation of concerns metrics. Such metrics are closely related to modularity, which influences comprehensibility, changeability, and independent development [17].

3 Classifications of Crosscutting Concerns

In the literature crosscutting concerns have been classified along three dimensions (homogeneous/heterogeneous) [18], (static/dynamic), and (basic/advanced) [19].

3.1 Homogeneous and Heterogeneous Crosscuts

A homogeneous Crosscut Extends a Program at Multiple Join Points by Adding the Same Piece of Code at Each Join Point.

¹ <http://www.jhotdraw.org>, version v.54b1

² <http://sourceforge.net/projects/ajhotdraw>

Table 1. Overview of the AspectJ programs analyzed [Sven Apel, Don Batory]

Name	LOC	Source	Description	Type
Tetris	1,030	Bekingse Inst. of Technology ^a	Implementation of the popular game	S
OAS	1,623	Lancaster University ^b	Online auction system	S
Preveyor	3,964	University of Toronto ^c	Main memory database system	R
AODP	3,995	University of Columbia ^d	AspectJ implementation of 23 design patterns	R
FACET	6,364	Washington University ^e	CORBA event channel implementation	S
ActiveAspect	6,664	University of British Columbia ^f	Crosscutting structure presentation tool	S
HealthWatcher	6,949	Lancaster University ^g	Web-based information system	S
AJHotDraw	22,104	open source project ^h	2D Graphics Framework	R
Hypercast	67,260	University of Virginia ⁱ	Protocol for multicast overlay networks	R
AJHSQLDB	75,556	University of Passau ^j	SQL relational database engine	R
Abacus	129,897	University of Toronto ^k	CORBA Middleware Framework	R

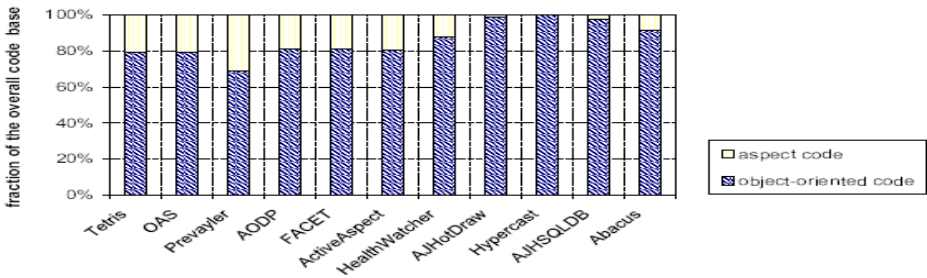


Fig. 1. Fractions of aspect code and object-oriented code of the overall code base. [Sven Apel, Don Batory]

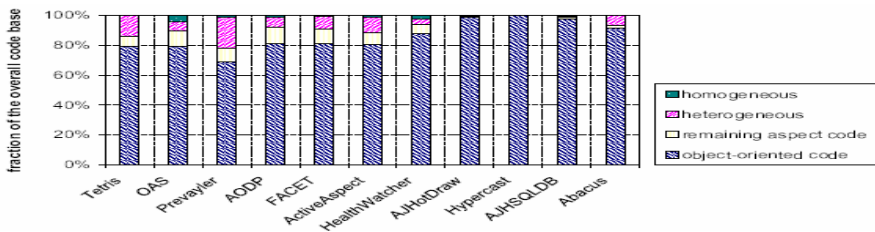


Fig. 2. Fractions of homogeneous and heterogeneous crosscuts of the overall code [Sven Apel, Don Batory]

A heterogeneous crosscut extends multiple join points each with a unique piece of code.

3.2 Static and Dynamic Crosscuts

A static crosscut extends the structure of a program statically, i.e., it adds new classes and interfaces and injects new fields, methods, and interfaces, etc. A dynamic crosscut affects the runtime control flow of a program and can be understood and defined

in terms of an event-based model: a dynamic crosscut executes additional code when predefined events occur during program execution [16].

3.3 Basic and Advanced Dynamic Crosscuts

The most primitive crosscut in AspectJ is a piece of advice that advises executions of a single method. Object-oriented languages express such advice as method extensions via subclassing (virtual classes or mixins), method overriding, and related mechanisms. Dynamic crosscutting mechanisms in AspectJ transcend object-oriented mechanisms when they effect events other than singleton method executions. Hence, we distinguish two classes of dynamic crosscuts, basic dynamic crosscuts and advanced dynamic crosscuts. Basic dynamic crosscuts: 1. affect executions of a single method, 2. access only runtime variables that are related to a method execution, i.e., arguments, result value, and enclosing object instance, and 3. affect a program control flow unconditionally [16].

3.4 Static and Dynamic Crosscuts (SDC)

The SDC metric determines the code fraction associated with static and dynamic crosscuts. That is, it counts the LOC of inter-type declarations (static crosscutting) and pieces of advice (dynamic crosscutting). Note that heterogeneous and homogeneous crosscuts can be either static or dynamic. The SDC metric tells us to what extent aspects crosscut the dynamic computation of a program or the static structure of a program. In AspectJ, the fraction of static and dynamic crosscuts are calculated by counting the LOC associated with inter-type declarations and pieces of advice and comparing them with the overall code base [16].

4 Code Metrics

To see how programmers use AspectJ, we define five metrics that distinguish the use of aspects in terms of the classifications discussed in the last section. For each metric, we count the number of lines of code (LOC) for different categories and determine the fraction of the program's source for each category. While using LOC (eliminating blank and comment lines) as a metric might be controversial (e.g., how are 'if' statements counted?), we will argue that the yielded statistics would be no different than, say, using a metric that counts statements. At the end, we compare just fractions of a program's code base associated with our categories. The essential results would remain valid [16].

4.1 Classes, Interfaces, and Aspects (CIA)

With the CIA metric we measure the fraction of classes, interfaces, and aspects of a program. It tells us whether aspects implement a significant or insignificant part of the code base (as opposed to classes and interfaces) [16].

We simply traverse all source files included in a given AspectJ project and count the LOC of aspects, classes, and interfaces. Upfront we eliminate blank lines and comments.

4.2 Heterogeneous and Homogeneous Crosscuts (HHC)

The HHC metric explores to what extent aspects implement heterogeneous and homogeneous crosscuts. Specifically, we determine the fractions of the LOC associated with advice and inter-type declarations (introductions) that are heterogeneous and homogeneous. The HHC metric tells us whether the implemented aspects take advantage of the advanced pattern-matching mechanisms of AspectJ.

We analyze each piece of advice and inter-type declaration: if its number of join points is greater than one it is a homogeneous crosscut; otherwise it is heterogeneous.

4.3 Code Replication Reduction (CRR)

Homogeneous advice and homogeneous inter-type declarations are useful for reducing code replication in a program. Suppose an aspect that advises 100 join points and executes at each join point 10 lines of code encapsulated in one piece of advice.

This aspect would reduce the code base by approximately 990 lines of code. In order to quantify this benefit, the CRR metric counts the LOC that could be reduced in an aspect-oriented version compared to an object-oriented equivalent [16].

We multiply the number of LOC of each homogeneous advice and inter-type declaration with the number of join points it affects (minus one).

5 Quantitative and Qualitative Metrics

The following metrics have been considered in the study [21], [22], [5], [9].

5.1 Concern Lines of Code (CLC)

This metric counts the number of lines of code whose main purpose is to contribute to the implementation of a concern (excluding comments, blank lines, and import statements) [23]. In the OO versions, CLC counts basically the method calls that support the considered crosscutting concerns. On the other hand, in the AO versions, the metric counts the size of the extracted aspects, in terms of lines of code (including point-cut descriptors, advice signatures, and advice implementations).

5.2 Concern Diffusion over Operations (CDO)

This metric counts the number of methods and advices whose main purpose is to contribute to the implementation of a concern and the number of other methods and advices that access them [5].

5.3 Concern Diffusion over Components (CDC)

This metric counts the number of classes and aspects whose main purpose is to contribute to the implementation of a concern and the number of other classes and aspects that access them [5].

The reason is clear, since the crosscutting code has been confined in a small number of aspects. From the presented quantitative assessment, we can expect that the

higher the quantification, the lower the lines of code related to the implementation of crosscutting concerns (CLC) and the diffusion of concerns over operations (CDO). However, **quantification** does not have an impact in the diffusion of concerns over components (CDC). In other words, **quantification** does not contribute to reduce the number of aspects required in the aspectization of a given system, but it contributes to reduce the number of advices and consequently the size of such aspects.

6 Quantification Metrics

In this section, we briefly discuss the application of quantification metrics in JHotDraw AspectJ System. JHotDraw is an object-oriented framework for 2D graphics. The system has been initially proposed as a design exercise to demonstrate the application of design patterns in a real application. Later, it has also been used as a case study in several papers about AOSD [24], [25]. Finally, some concerns of the system have been refactored to aspects by Marin et al., [22], [25] leading to a version called AJHotDraw. In this example, we have calculated the proposed QD and SR metrics for three methods related to different crosscutting concerns: `fireSelectionChanged` (that is part of the figure selection concern), `setUndoActivity` (that is part of the undo concern), and `checkDamaged` (that is part of the command concern). We have chosen such methods by comparing the OO and AO versions of the system. In the AO version, they are called by code that resides in advices.

We have used a prototype tool, called ConcernMetrics[9] and AOPMetric [26] that estimates both conventional separation of concern metrics and the quantification metrics proposed without requiring developers to implement aspects. Our main motivation is to understand and have the same opinion, AOP quantification mechanism helps developers and maintainers to decide in a cost-effective way if it is worthwhile to use aspects in their systems [9].

Table 2 presents JHotDraw's [9] QD and SR values. As can be observed, for two methods we have found high values for QD: `fireSelectionChanged` (QD = 1) and `setUndoActivity` (QD = 0.92). On the other hand, 15 advices have been implemented to modularize the existent 28 calls to `checkDamaged`. Therefore, QD = 0.48, which shows that intermediary QD values can be observed in practice (since in the other examples the values are usually in the extremes of the scale).

Table 2. JHotDraw's QD and SR [Marco Tulio Valente et al]

Concern	<i>jps</i>	<i>adv</i>	SR	QD
<code>fireSelectionChanged()</code>	4	1	3	1
<code>setUndoActivity(Undoable)</code>	13	2	11	0.92
<code>checkDamaged()</code>	28	15	13	0.48

Table 3 presents the SR and QD values calculated from the AJHotDraw—JHotDraw’s [9], [25], [23], [26] AO version—and the values estimated by the ConcernMetrics directly from the OO version of the system. The differences observed in the results are explained as follows. The calls to fireSelection could have been confined in a single advice (instead AJHotDraw’s designers have chosen to implement them in two advices). In the case of setUndoActivity, AJHotDraw’s designers have left four calls scattered in the OO code (i.e., such calls have not been extracted to aspects). Finally, from the 28 calls to checkDamaged that exist in the OO code, only 12 calls have been moved to aspects [9].

Table 3. SR and QD for JHotDraw (AO = value measured for the AO version; CM = value estimated by the ConcernMetrics) [Marco Tulio Valente et al]

Concern	AO		CM	
	SR	QD	SR	QD
1 fireSelectionChanged()	2	0.67	3	1
2 setUndoActivity(Undoable)	1	0.13	11	0.92
3 checkDamaged()	12	1	13	0.48

7 Conclusion

In this paper we have identified AOP with the ability to assert quantified statements over programs written by oblivious programmers. We have provided insight that quantification is the key mechanism to abstract out computations associated to cross-cutting concerns. Quantification allows developers to implement in a single advice code required in many static locations of object-oriented systems. Therefore, quantification directly tackles the code scattering and tangling problems that typically characterize crosscutting concerns. In order to better assess the benefits of quantification, we have identified two more metrics: quantification degree (QD) and scattering reduction (SR) [9]. The best tool available as per literature is ConcernMetrics and AOPMetrics tool [26] that estimates AOP-related metrics directly from the object-oriented code. A comparison of development aspects and aspects actually deployed could be a topic of further research.

Acknowledgment

We place on records and wish to thank the authors M.T. Valente, C Couto et al., Department of Computer Science, UFMG, Belo Horizonte, Brazil, Sven Apel, Department of Informatics and Mathematics, University of Passau, Germany for their valuable contributions to this work and providing useful research on benefits of on AO Systems quantification.

References

1. http://www.ercim.eu/publication/Ercim_News/enw58/mens.html
2. Steimann, F.: The paradoxical success of aspect-oriented programming. In: 21st Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA), pp. 481–497 (2006)
3. Wand, M.: Understanding aspects: extended abstract. In: 8th International Conference on Functional Programming (ICFP), pp. 299–300. ACM, New York (2003)
4. Apel, S.: How AspectJ is used: an analysis of eleven AspectJ programs. *J. Object Technol.* 9(1), 117–142 (2010)
5. Sant’Anna, C., Garcia, A., Chavez, C., Lucena, C., von Staa, A.: On the reuse and maintenance of aspect-oriented software: an assessment framework. In: 17th Brazilian Symposium on Software Engineering (SBES), pp. 19–34 (2003)
6. Wand, M.: Understanding aspects: extended abstract. In: 8th International Conference on Functional Programming (ICFP), pp. 299–300. ACM, New York (2003)
7. Kulesza, U., Sant’Anna, C., Garcia, A., Coelho, R., von Staa, A., Luce Na, C.: Quantifying the Effects of Aspect-Oriented Programming: A Maintenance Study. In: Proc. Int’l. Conf. Software Maintenance (2006)
8. Parnas, D.L.: On the criteria to be used in decomposing systems into modules. *Commun ACM* 15(12), 1053–1058 (1972)
9. Valente, M.T., et al.: On the benefits of quantification in Aspect J systems. *J. Braz Comput. Soc.*, doi: 10.1007/s13173-010-0008-0
10. Zhao, J., Xu, B.: Measuring Aspect Cohesion. In: Wermelinger, M., Margaria-Steffen, T. (eds.) FASE 2004. LNCS, vol. 2984, pp. 54–68. Springer, Heidelberg (2004)
11. Zhang, C., Jacobsen, H.-A.: Resolving Feature Convolution in Middleware Systems. In: Proc. Int’l. Conf. Object-Oriented Programming, Systems, Languages, and Applications (2004)
12. Garcia, A., Sant’Anna, C., Figueiredo, E., Kulesza, U., Lucena, C.J.P., von Staa, A.: Modularizing Design Patterns with Aspects: A Quantitative Study. In: Rashid, A., Liu, Y. (eds.) Transactions on Aspect-Oriented Software Development I. LNCS, vol. 3880, pp. 36–74. Springer, Heidelberg (2006)
13. Kulesza, U., Sant’Anna, C., Garcia, A., Coelho, R., von Staa, A., Luce Na, C.: Quantifying the Effects of Aspect-Oriented Programming: A Maintenance Study. In: Proc. Int’l. Conf. Software Maintenance (2006)
14. Gelinas, J.F., Badri, M., Badri, L.: A Cohesion Measure for Aspects. *J. Object Technology* 5(7) (2006)
15. Lopez-Herrejon, R.E., Apel, S.: Measuring and Characterizing Crosscutting in Aspect-Based Programs: Basic Metrics and Case Studies. In: Dwyer, M.B., Lopes, A. (eds.) FASE 2007. LNCS, vol. 4422, pp. 423–437. Springer, Heidelberg (2007)
16. Apel, S.: How AspectJ is used: an analysis of eleven AspectJ programs. *J. Object Technol.* 9(1), 117–142 (2010)
17. Parnas, D.L.: On the criteria to be used in decomposing systems into modules. *Commun. ACM* 15(12), 1053–1058 (1972)
18. Colyer, A., Rashid, A., Blair, G.: On the Separation of Concerns in Program Families. Technical Report COMP-001-2004, Computing Department, Lancaster University (2004)
19. Apel, S., Leich, T., Saake, G.: Aspectual Feature Modules. *IEEE Trans. Software Engineering* 34(2) (2008)

20. Garcia, A., Sant'Anna, C., Figueiredo, E., Kulesza, U., de Lucena, C.J.P., von, S.: Modularizing design patterns with aspects: a quantitative study. In: 4th International Conference on Aspect Oriented Software Development (AOSD), pp. 3–14 (2005)
21. Greenwood, P., Bartolomei, T.T., Figueiredo, E., Dósea, M., Garcia, A.F., Cacho, N., Sant'Anna, C., Soares, S., Borba, P., Kulesza, U., Rashid, A.: On the impact of aspectual decompositions on design stability: An empirical study. In: Bateni, M. (ed.) ECOOP 2007. LNCS, vol. 4609, pp. 176–200. Springer, Heidelberg (2007)
22. Marin, M., van Deursen, A., Moonen, L., van der Rijst, R.: An integrated crosscutting concern migration strategy and its Semiautomated application to JHotDraw. *Autom. Softw. Eng.* 16(2), 323–356 (2009)
23. Binkley, D., Ceccato, M., Harman, M., Ricca, F., Tonella, P.: Automated refactoring of object oriented code into aspects. In: 21st IEEE International Conference on Software Maintenance (ICSM), pp. 27–36 (2005)
24. Eaddy, M., Zimmermann, T., Sherwood, K.D., Garg, V., Murphy Gail, C., Nagappan, N., Aho, A.V.: Do crosscutting concerns cause defects? *IEEE Trans. Softw. Eng.* 34(4), 497–515 (2008)
25. Marin, M., van Deursen, A., Moonen, L.: Identifying crosscutting concerns using fan-in analysis. *ACM Trans. Softw. Eng. Methodol.* 17(1) (2007)
26. <http://aopmetrics.tigris.org/>

Empirical Measurements on the Convergence Nature of Differential Evolution Variants

G. Jeyakumar¹ and C. Shanmugavelayutham²

^{1,2} Assistant Professor

Department of Computer Science and Engineering

Amrita School of Engineering

Amrita Vishwa VidyaPeetham

Coimbatore, Tamil Nadu, India

g_jeyakumar@cb.amrita.edu,

cs_velayutham@cb.amrita.edu

Abstract. In this paper, we present an empirical study on convergence nature of Differential Evolution (DE) variants to solve unconstrained global optimization problems. The aim is to identify the convergence behavior of DE variants and compare. We have chosen fourteen benchmark functions grouped by feature: unimodal separable, unimodal nonseparable, multimodal separable and multimodal nonseparable. Fourteen variants of DE were implemented and tested on these problems for dimensions of 30. The variants are well compared by their Convergence Speed, Quality Measure and Population Convergence Measure.

Keywords: Differential Evolution, Variants, Global Optimization, Convergence, Population Variance.

1 Introduction

Evolutionary algorithms (EA) have been widely used to solve optimization problems. Differential Evolution [1] is an EA proposed to solve optimization problems, mainly to continuous search spaces. The DE algorithm has been successfully applied to many global optimization problems [2]. As traditional EAs, several optimization problems have been successfully solved by using DE [3]. It shows superior performance in both widely used benchmark functions and real-world application [4, 5]. DE shares similarities with traditional EAs. As in other EAs, two main processes are the perturbation process (crossover and mutation) which ensures the exploration of the search space and the selection process which ensures the exploitation properties of the algorithm. Both perturbation and the selection process are simpler in DE, the perturbation of a population element is done by probabilistically replacing it with an offspring obtained by adding to a randomly selected element a perturbation proportional with the difference between other two randomly selected elements. The selection is done by one to one competition between the parent and its offspring. The structure of DE algorithm is presented in Figure. 1.

Based on different strategies followed for perturbation, there exists various DE variants, they differ in the way how the solution is generated. With seven commonly

```

Population Initialization  $X(0) \leftarrow \{x_1(0), \dots, x_{NP}(0)\}$ 
 $g \leftarrow 0$ 
Compute  $\{f(x_1(g)), \dots, f(x_{NP}(g))\}$ 
while the stopping condition is false do
  for  $i = 1$  to NP do
    MutantVector:  $y_i \leftarrow \text{generatemutant}(X(g))$ 
    TrialVector:  $z_i \leftarrow \text{crossover}(x_i(g), y_i)$ 
    if  $f(z_i) < f(x_i(g))$  then
       $x_i(g+1) \leftarrow z_i$ 
    else
       $x_i(g+1) \leftarrow x_i(g)$ 
    end if
  end for
   $g \leftarrow g+1$ 
  Compute  $\{f(x_1(g)), \dots, f(x_{NP}(g))\}$ 
end while

```

used differential mutation strategies, and two crossover schemes (binomial and exponential), we get fourteen possible variants of DE (as listed in Table 2). In this paper, an empirical convergence analysis of these 14 variants has been attempted.

The remainder of the paper is organized as follows. After a brief review of the related work in Section 2, Section 3 details the design of experiments and describes about the numerical tests in our study, Section 4 discusses the simulation results and finally Section 5 concludes the paper.

Fig. 1. Description of DE algorithm

2 Related Works

Menzura-Montes et. al. [6] empirically compared the performance of eight DE variants. He used convergence measure to identify the competitiveness of the variants. The study concluded *rand/1/bin*, *best/1/bin*, *current-to-rand/1/bin* and *rand/2/dir* as the most competitive variants. However, the potential variants like *best/2/**, *rand-to-best/1/** and *rand/2/** were not considered in their study.

Daniela Zaharie [7] provides theoretical insights on explorative power of Differential Evolutionary algorithms, she describes an expression as a measure of the explorative power. In her results, the evolution of population variance for *rand/1/bin* is measured for two test functions. Control of diversity and associated parameter tuning are discussed in [8, 9].

Hans-Georg Beyer [10] analyzed how the ES/EP-like algorithms perform the evolutionary search in the real-valued N-dimensional spaces. He described the search behavior as the antagonism of exploitation and exploration.

3 Design of Experiments

In this paper, we investigate the convergence nature of DE variants and compare them by implementing on a set of benchmark. We have chosen fourteen test functions [6, 11], of dimensionality 30, grouped by the feature - unimodal separable, unimodal nonseparable, multimodal separable and multimodal nonseparable. The details of the benchmark problems are described in Table 1.

All the test functions have an optimum value at zero except *f08*. In order to show the similar results, the description of *f08* is adjusted to have its optimum value at zero by just adding the value (12569.486618164879) [6]. The parameters for all the DE

variants were: population size NP = 60 and maximum number of generations = 3000. The variants will stop before the maximum number of generations is reached only if the tolerance error (which has been fixed as an error value of 1×10^{-12}) is obtained. Following [6, 12], we defined a range for the scaling factor, $F \in [0.3, 0.9]$. We use the same value for K as F.

Table 1. Details of the test functions used in the experiment

Functions and Ranges	Functions and Ranges
$f01 : f_{Sp}(x) = \sum_{i=1}^{30} x_i^2 ; -100 \leq x_i \leq 100$	$f08 : f_{Sch}(x) = \sum_{i=1}^{30} (x_i \sin(\sqrt{ x_i })) ; -500 \leq x_i \leq 500$
$f02 : f_{Sch}(x) = \sum_{i=1}^{30} x_i + \prod_{i=1}^{30} x_i ; -10 \leq x_i \leq 10$	$f09 : f_{Ras}(x) = \sum_{i=1}^{30} [x_i^2 - 10 \cos(2\pi x_i) + 10] ; -5.12 \leq x_i \leq 5.12$
$f03 : f_{schDS}(x) = \sum_{i=1}^{30} (\sum_{j=1}^i x_j)^2 ; -100 \leq x_i \leq 100 ;$	$f10 : f_{Ack}(x) = 20 + e - 20 \exp\left(-0.2 \sqrt{\frac{1}{p} \sum_{i=2}^p x_i^2}\right) - \exp\left(\frac{1}{p} \sum_{i=1}^p \cos(2\pi x_i)\right) ; -30 \leq x_i \leq 30$
$f04 : f_{sch}(x) = \max_i \{ x_i , 1 \leq i \leq 30\} ; -100 \leq x_i \leq 100$	$f11 : f_{Gri}(x) = \frac{1}{4000} \sum_{i=1}^{30} x_i^2 - \prod_{i=1}^{30} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 ; -600 \leq x_i \leq 600$
$f05 : f_{Ros}(x) = \sum_{i=1}^{29} 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 ; -30 \leq x_i \leq 30$	$f12 : f_{GPF12} = \frac{\pi}{30} \{10 \sin^2(\pi y_1) + \sum_{i=1}^{30} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^{30} u(x_i, 10, 100, 4) ; -50 \leq x_i \leq 50$
$f06 : f_{St}(x) = \sum_{i=1}^{30} (x_i + 0.5)^2 ; -1.28 \leq x_i \leq 1.28$	$f13 : f_{GPF13} = 0.1 \{ \sin^2(\pi 3x_1) + \sum_{i=1}^{29} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + (x_n - 1)^2 \} [1 + \sin^2(2\pi x_{30})] + \sum_{i=1}^{30} u(x_i, 10, 100, 4) ; -50 \leq x_i \leq 50$
$f07 : f_{QF}(x) = \sum_{i=1}^{30} i x_i^4 + \text{randon}\{0, 1\} ; -1.28 \leq x_i \leq 1.28$	$f14 : f_{Boh} = x_i^2 + 2x_{i+1}^2 - 0.3 \cos(3\pi x_i) - 0.4 \cos(4\pi x_{i+1}) + 0.7 ; -100 \leq x_i \leq 100$

The crossover rate, CR, was tuned for each variant-test function combination. Eleven different values for the CR viz. {0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 and 1.0} were used to conduct a bootstrap test, in order to determine the confidence interval for the mean objective function value. The CR value corresponding to the best confidence interval was chosen to be used in our experiment. The fourteen variants of DE along with the CR values for each test function are presented in Table 2.

As EA’s are stochastic in nature, 100 independent runs were performed per variant per test function. For the sake of performance analysis among the variants, we present the mean objective function values (MOV), Convergence Speed [6], Quality Measure (Q-Measure) [13] and Probability of Convergence Measure (P-Measure) [13] for each variant-test function combination.

Convergence Speed is used to detect which variant is most competitive. To measure the convergence speed, we calculated the mean percentage out of the total 1,80,000 function evaluations required by each of the variant to reach its best objective function value, for all the 100 independent runs.

Table 2. CR Value Measured for Each Pair of Variant-Function

Sno	Variant	<i>f01/f08</i>	<i>f02/f09</i>	<i>f03/f10</i>	<i>f04/f11</i>	<i>f05/f12</i>	<i>f06/f13</i>	<i>f07/f14</i>
1	<i>rand/1/bin</i>	0.9/0.5	0.2/0.1	0.9/0.9	0.5/0.1	0.9/0.1	0.2/0.1	0.8/0.1
2	<i>rand/1/exp</i>	0.9/0.0	0.9/0.9	0.9/0.9	0.9/0.9	0.9/0.9	0.9/0.9	0.9/0.9
3	<i>best/1/bin</i>	0.1/0.1	0.1/0.1	0.5/0.1	0.2/0.1	0.8/0.3	0.1/0.8	0.7/0.1
4	<i>best/1/exp</i>	0.9/0.7	0.8/0.9	0.9/0.8	0.9/0.8	0.8/0.9	0.8/0.8	0.9/0.8
5	<i>rand/2/bin</i>	0.3/0.2	0.1/0.1	0.9/0.1	0.2/0.1	0.9/0.1	0.2/0.1	0.9/0.1
6	<i>rand/2/exp</i>	0.9/0.3	0.9/0.9	0.9/0.9	0.9/0.9	0.9/0.9	0.9/0.9	0.9/0.9
7	<i>best/2/bin</i>	0.1/0.7	0.3/0.1	0.7/0.4	0.2/0.1	0.6/0.1	0.1/0.1	0.5/0.1
8	<i>best/2/exp</i>	0.9/0.3	0.9/0.9	0.9/0.9	0.9/0.9	0.9/0.9	0.9/0.9	0.9/0.9
9	<i>current-to-rand/1/bin</i>	0.5/0.4	0.1/0.1	0.9/0.1	0.2/0.1	0.1/0.2	0.1/0.3	0.2/0.1
10	<i>current-to-rand/1/exp</i>	0.9/0.3	0.9/0.9	0.9/0.9	0.9/0.9	0.9/0.9	0.9/0.9	0.9/0.9
11	<i>current-to-best/1/bin</i>	0.2/0.8	0.1/0.1	0.9/0.1	0.2/0.2	0.1/0.2	0.3/0.1	0.2/0.1
12	<i>current-to-best/1/exp</i>	0.9/0.1	0.9/0.9	0.9/0.9	0.9/0.9	0.9/0.9	0.9/0.9	0.9/0.9
13	<i>rand-to-best/1/bin</i>	0.1/0.8	0.1/0.1	0.9/0.9	0.4/0.1	0.8/0.1	0.4/0.2	0.8/0.1
14	<i>rand-to-best/1/exp</i>	0.9/0.4	0.9/0.9	0.9/0.9	0.9/0.9	0.9/0.9	0.9/0.9	0.9/0.9

We used Q-Measure and P-Measure to study the convergence nature of DE Variants. Quality measure or simply Q-Measure is an empirical measure of the algorithm's convergence. In our experiment, it is used to study the behavior of our DE variants. The formula of Q-measure is $Q_m = C / P_c$ (where, P_c is Probability of convergence and C is Convergence Measure). The Convergence Measure(C) is calculated as $C = \text{Sum}E_j / nc$ (Where, nc is number of successful runs and $\text{Sum}E_j$ is total number of function evaluations taken for all the successful runs). The value for P_c is calculated as $P_c = nc/nt$ (where, nt is the total number of runs).

The convergence of a population is a measure that allows us to observe the convergence rate from the point of view of variables. The population convergence or simply P-measure (P_m) is a radius of a population. It can also be stated as the Euclidean distance between the center of a population and the individual farthest from it. The center of the population (O_p) is calculated as the average vector of all individuals, $O_p = \text{Sum of Indi} / NP$. The P-measure is calculated as $P_m = \max \| \text{indi} - O_p \| E$, for $i = 1, \dots, NP$.

4 Results and Discussion

The mean objective function values obtained for the unimodal separable functions: *f01*, *f02*, *f04*, *f06* and *f07*, and the unimodal nonseparable function *f03* are presented in Table 3. The results shows that best performance were provided by *rand-to-best/1/bin*, *rand/1/bin*, *best/2/bin* and *rand/2/bin* variants for the unimodal functions. *best/1/**, *current-to-rand/1/exp* and *current-to-best/1/exp* were the poorly performing variants.

Table 3. MOV Obtained for Unimodal Functions

Variant	f01	f02	f04	f06	f07	f03
rand/1/bin	0.00	0.00	0.00	0.02	0.00	0.07
rand/1/exp	0.00	0.00	3.76	0.00	0.02	0.31
best/1/bin	457.25	0.14	1.96	437.25	0.09	13.27
best/1/exp	583.79	4.05	37.36	591.85	0.06	57.39
rand/2/bin	0.00	0.00	0.06	0.00	0.01	1.64
rand/2/exp	0.00	0.02	32.90	0.00	0.05	269.86
best/2/bin	0.00	0.00	0.00	0.07	0.00	0.00
best/2/exp	0.00	0.00	0.05	0.39	0.01	0.00
current-to-rand/1/bin	0.00	0.02	3.68	0.03	0.04	3210.36
current-to-rand/1/exp	24.29	44.22	57.52	43.07	0.27	3110.90
current-to-best/1/bin	0.00	0.02	3.71	0.00	0.04	3444.00
current-to-best/1/exp	24.37	45.04	56.67	41.95	0.26	2972.62
rand-to-best/1/bin	0.00	0.00	0.00	0.00	0.00	0.07
rand-to-best/1/exp	0.00	0.00	3.38	0.00	0.01	0.20

Table 4 displays the simulation results for the multimodal separable functions: *f08*, *f09* and *f14*, and for the multimodal non-separable functions *f05*, *f10*, *f11*, *f12* and *f13*. In case of multimodal separable problems, the best performance was shown by *rand/1/bin*, *rand-to-best/1/bin* and *rand/2/bin* variants once again as in the case of unimodal separable problems. Similarly, the variants *current-to-rand/1/exp* and *current-to-best/1/exp* consistently showed poor performance. In the case of multimodal nonseparable problems, function *f05* and *f10* were not solved by any variants. In this case of *f11*, *f12* and *f13*, the best performing variants were *rand-to-best/1/bin*, *rand/1/bin*, *rand/2/bin* and *best/2/bin*, once again along with *current-to-rand/1/bin* and *rand-to-best/1/bin*. *current-to-rand/1/exp* and *current-to-best/1/exp* were the poorly performing variants along with *best/1/** variants.

Table 4. MOV Obtained for Multimodal Functions

Variant	f08	f09	f14	f05	f10	f11	f12	f13
rand/1/bin	0.13	0.00	0.00	21.99	0.09	0.00	0.00	0.00
rand/1/exp	0.10	47.93	0.00	25.48	0.09	0.05	0.00	0.00
best/1/bin	0.00	4.33	12.93	585899.88	3.58	3.72	15.78	973097.03
best/1/exp	0.01	50.74	32.18	64543.84	6.09	5.91	131448.66	154434.94
rand/2/bin	0.22	0.00	0.00	19.01	0.09	0.00	0.00	0.00
rand/2/exp	0.27	101.38	0.01	2741.32	0.01	0.21	0.00	0.01
best/2/bin	0.17	0.69	0.12	2.32	0.09	0.00	0.00	0.00
best/2/exp	0.08	80.63	2.53	1.12	0.83	0.03	0.14	0.00
current-to-rand/1/bin	0.14	37.75	0.00	52.81	0.01	0.00	0.00	0.00
current-to-rand/1/exp	0.12	235.14	18.35	199243.32	13.83	1.21	10.89	24.11
current-to-best/1/bin	0.19	37.04	0.00	56.91	0.01	0.00	0.00	0.00
current-to-best/1/exp	0.10	232.80	18.21	119685.68	13.69	1.21	10.37	23.04
rand-to-best/1/bin	0.22	0.00	0.00	17.37	0.09	0.00	0.00	0.00
rand-to-best/1/exp	0.12	48.09	0.00	24.54	0.09	0.05	0.00	0.00

Based on the overall results in Table 3 and 4 the most competitive variants were *rand-to-best/1/bin*, *best/2/bin* and *rand/1/bin*. The variants *rand/2/bin* and *best/2/exp* also showed good performance consistently. On the other hand, the worst overall

performance was consistently displayed by variants *current-to-best/1/exp* and *current-to-rand/1/exp*. It is worth noting that binomial recombination showed a better performance over the exponential recombination. Next in our experiment, we measured the Convergence Speed as the mean percentage of total number of function evaluations required by each of the variant, for 100 runs, to reach their best objective function value. Table 5 shows the Convergence Speed of the variants, ordered by function groups.

Table 5. Percentage of the total number of function evaluations taken by each variant for 100 runs (grouped by the function class). The lowest percentage is marked with “*”.

Variant	Function	f01	f02	f04	f06	f07	f03	f08	f09	f14	f05	f10	f11	f12	f13
1.DE/rand/1/bin		40.93*	56.45	100	10.89*	100	100	100	65.5	42.13	100	100	46.88	38.89*	41.38
2.DE/rand/1/exp		100	100	100	39.34	100	100	100	100	100	100	100	100	100	100
3.DE/best/1/bin		100	100	100	100	100	100	100	100	100	100	100	100	100	100
4.DE/best/1/exp		100	100	100	100	100	100	100	100	100	100	100	100	100	100
5.DE/rand/2/bin		70.05	90.59	100	17.78	100	100	100	100	59.84	100	100	70.02	56.07	58.62
6.DE/rand/2/exp		100	100	100	81.64	100	100	100	100	100	100	100	100	100	100
7.DE/best/2/bin		42.67	48.9*	100	12.47	100	100	100	87.29	48.6	99.97*	100	49.6	40.05	41.23
8.DE/best/2/exp		73.42	100	100	51.93	100	99.85*	100	100	98.35	100	100	88.94	80.43	81.42
9.DE/current-to-rand/1/bin		100	100	100	40.13	100	100	100	100	100	100	100	100	100	100
10.DE/current-to-rand/1/exp		100	100	100	100	100	100	100	100	100	100	100	100	100	100
11.DE/current-to-best/1/bin		100	100	100	50.84	100	100	100	100	100	100	100	100	100	100
12.DE/current-to-best/1/exp		100	100	100	100	100	100	100	100	100	100	100	100	100	100
13.DE/rand-to-best/1/bin		43.11	63.57	100	11.45	100	100	100	65.1*	42.09*	100	100	46.6*	38.94	38.34*
14.DE/rand-to-best/1/exp		100	100	100	39.11	100	100	100	100	100	100	100	100	100	100

The results show that for the unimodal functions the top four variants with the fastest convergence are *rand/1/bin*, *rand/2/bin*, *best/2/bin* and *rand-to-best/1/bin*. The variants *best/1/bin*, *best/1/exp*, *current-to-rand/1/exp* and *current-to-best/1/exp* are slow in convergence. For *f03*, *best/2/exp* is comparatively faster than other variants by 0.05%. For the multimodal functions variants with the fastest convergence are the same variants as in the case of unimodal functions. The results suggest that, the variants *rand-to-best/1/bin* and *rand/1/bin* are out performing others by its convergence speed and the variants *best/1/bin*, *best/1/exp*, *current-to-rand/1/exp* and *current-to-best/1/exp* are slow in convergence.

The values of Q_m are presented in the Table 6. For the unimodal functions, the top four variants with least Q_m values are *rand/1/bin*, *best/2/bin*, *rand-to-best/1/bin* and *rand/2/bin*. The variants *best/1/exp*, *current-to-rand/1/exp* and *current-to-best/1/exp* could not provide any successful run. This suggests the influence of binomial recombination in DE variants. In case of multimodal functions also the similar trend was observed.

The population convergence (P_m) is used as a measure to compare the convergence nature of the variants. We have done this experiment for all the Variant-Function combination. It is presented for *f01* and *f09* by the variants *rand/1/bin* and *best/1/bin*, because these are the two variants have different convergence behavior in our study. The results, Table 7, show that for the function *f01*, the variant *best/1/bin* gives both the best and worst run as unsuccessful one, due to its premature convergence. And the variant *rand/1/bin* gives both runs as successful run, because it could maintain the diversity in population. For the function *f03*, it is observed that the *rand/1/bin* reaches the global optimum, but *best/1/bin* could not converge to global optimum due to stagnation problem.

Table 6. Q-measure Measurement for the Variants

Variant	Function	f01	f02	f03	f04	f05	f06	f07	f08	f09
1.DE/rand/1/bin		73686.6	10162080	13140000	18000000	-	1922580	10800000	720000	11790660
2.DE/rand/1/exp		18000000	18000000	720000	-	-	7081860	-	1260000	-
3.DE/best/1/bin		540000	7200000	15480000	14220000	-	-	3960000	15840000	540000
4.DE/best/1/exp		-	-	10440000	-	-	-	-	15300000	-
5.DE/rand/2/bin		12608200	16305480	-	-	-	3201060	360000	180000	18000000
6.DE/rand/2/exp		10980000	-	-	-	-	14695800	-	360000	-
7.DE/best/2/bin		7679760	8819160	18000000	18000000	6835320	2135280	13500000	180000	6171360
8.DE/best/2/exp		13214760	18000000	18000000	180000	5220000	3767640	3600000	3060000	-
9.DE/current-to-rand/1/bin		18000000	-	-	-	-	7222800	-	360000	-
10.DE/current-to-rand/1/exp		-	-	-	-	-	-	-	540000	-
11.DE/current-to-best/1/bin		18000000	-	-	-	-	9150660	-	540000	-
12.DE/current-to-best/1/exp		-	-	-	-	-	-	-	900000	-
13.DE/rand-to-best/1/bin		7759500	11442000	14220000	18000000	-	2061060	10800000	-	11718600
14.DE/rand-to-best/1/exp		18000000	-	1800000	-	-	7039200	-	1080000	-
Variant	Function	f10	f11	f12	f13	f14	SumEj	C	Qm=C/Pc	
1.DE/rand/1/bin		-	8437680	7000500	7448880	7584180	97080246.6	93797.34	1268.76	
2.DE/rand/1/exp		-	12240000	18000000	18000000	18000000	111301860	163920.27	3379.8	
3.DE/best/1/bin		-	180000	-	-	-	57960000	180000	7826.09	
4.DE/best/1/exp		-	-	-	-	-	25740000	180000	17622.38	
5.DE/rand/2/bin		-	12603420	10092120	10551840	10772040	94674160	117900.57	2055.55	
6.DE/rand/2/exp		11520000	540000	18000000	9000000	4680000	81295800	200235.96	6904.69	
7.DE/best/2/bin		-	8927340	7028940	7422060	6767700	111466920	106769.08	1431.77	
8.DE/best/2/exp		-	5929380	8896560	9076380	956880	86661600	142770.35	3292.89	
9.DE/current-to-rand/1/bin		10080000	17280000	17280000	18000000	18000000	106222800	163419.69	3519.81	
10.DE/current-to-rand/1/exp		-	-	-	-	-	540000	180000	840000	
11.DE/current-to-best/1/bin		10080000	17280000	16380000	18000000	18000000	107430660	166301.33	3604.05	
12.DE/current-to-best/1/exp		-	-	-	-	-	900000	180000	504000	
13.DE/rand-to-best/1/bin		-	8388600	7008720	6901200	7576620	105876300	101902.12	1373.08	
14.DE/rand-to-best/1/exp		-	12420000	18000000	18000000	18000000	94339200	161263.59	3859.3	

Table 7. P_m Measurement for $f0I$

Best Run – Successful Run							Best Run – Unsuccessful Run						
Run	G	ObjValue	Mean	Stddev	Variance	P_m	Run	G	ObjValue	Mean	Stddev	Variance	P_m
9	0	57466.93	98460.94	18822.72	3233.92	396.43	8	0	63579.26	97917.25	16186.32	3210.66	361.25
9	100	4319.44	6929.9	1255.32	194.56	100.58	8	100	733.05	1594.22	639.62	46.72	60.21
9	200	115.76	170.93	30.98	4.14	14.35	8	200	6.76	13.13	4.36	0.38	4.73
9	300	1.47	2.48	0.4	0.06	1.82	8	292	0.09	0.17	0.07	0	0.64
9	369	0.13	0.19	0.03	0	0.49	8	351	0.02	0.02	0	0	0.15
9	453	0	0.01	0	0	0.1	8	510	0.01	0.01	0	0	0
9	601	0	0	0	0	0	8	600	0.01	0.01	0	0	0
9	900	0	0	0	0	0	8	900	0.01	0.01	0	0	0
9	1084	0	0	0	0	0	8	1200	0.01	0.01	0	0	0
							8	1500	0.01	0.01	0	0	0
							8	1800	0.01	0.01	0	0	0
							8	2100	0.01	0.01	0	0	0
							8	2400	0.01	0.01	0	0	0
							8	2999	0.01	0.01	0	0	0
Worst Run – Successful Run							Worst Run – Unsuccessful Run						
Run	G	ObjValue	Mean	Stddev	Variance	P_m	Run	G	ObjValue	Mean	Stddev	Variance	P_m
34	0	63161.05	97475.45	18002.08	3184.79	378.27	39	0	56650.36	98206.15	17038.53	3233.84	367.82
34	100	2740.98	4292.98	538.96	105.59	70.93	39	100	3338.62	4554.26	1038.42	61.38	76.01
34	200	134.65	188.87	29.6	3.41	13.96	39	200	2523.86	2536.92	7.8	0.58	6.58
34	300	7.16	8.99	0.93	0.12	2.51	39	300	2517.97	2518.11	0.08	0.01	0.68
34	417	0.2	0.27	0.04	0	0.56	39	313	2517.9	2517.99	0.05	0	0.54
34	565	0	0.01	0	0	0.09	39	365	2517.84	2517.84	0	0	0.17
34	700	0	0	0	0	0.02	39	534	2517.83	2517.83	0	0	0
34	770	0	0	0	0	0	39	900	2517.83	2517.83	0	0	0
34	900	0	0	0	0	0	39	1200	2517.83	2517.83	0	0	0
34	1200	0	0	0	0	0	39	1500	2517.83	2517.83	0	0	0
34	1514	0	0	0	0	0	39	1800	2517.83	2517.83	0	0	0
							39	2100	2517.83	2517.83	0	0	0
							39	2400	2517.83	2517.83	0	0	0
							39	2700	2517.83	2517.83	0	0	0
							39	2999	2517.83	2517.83	0	0	0

For the function $f09$, Table 8, both the variants could reach the global optimum at their best run, but $rand/1/bin$ is faster in convergence, it reaches with less number of function evaluations. And the variant $rand/1/bin$ could reach the global optimum in its worst run also. But the variant $best/1/bin$ falls in premature convergence in its worst run. Similarly for the function $f05$ the variant $best/1/bin$ falls in premature convergence, even at its best run. But the variant “DE/ $rand/1/bin$ ” could reach the global optimum in its best run.

The results suggest that the variant $rand/1/bin$ is good in convergence. It is observed from various results that, even though selection decreases the population variance the variation operators of $rand/1/bin$ could balance it by exploring the population in all the directions. This is due to the randomness property of the variant, which is used for selecting the candidates for perturbation. On the other hand, the variant $best/1/bin$ falls in premature convergence (in most of the cases), the identified reason for this behavior is that the variation operators of this variant is not balancing the effect made by the selection operator in the population diversity.

In $best/1/bin$, the candidates for mutation operation is selected only in the direction from the best candidate of the current population, this makes the exploration operation to lose its power, and subsequently vanishes the population diversity soon. This effect leads to premature convergence.

Table 8. P_m Measurement for $f09$

DE/ $rand/1/bin$							DE/ $best/1/bin$						
Best Run – Successful Run							Best Run – Successful Run						
Run	G	ObjValue	Mean	Stddev	Variance	Pm	Run	G	ObjValue	Mean	Stddev	Variance	Pm
37	0	390.49	555.73	59.33	8.46	18.6	83	0	437.19	569.66	60.82	8.98	19.03
37	100	194.04	262.05	37.1	3.08	12.61	83	100	150.8	199.89	23.28	2.06	10.47
37	230	104.07	155.73	22.19	1.7	9.51	83	200	78.07	120.79	16.11	1.23	8.85
37	310	82.5	114.17	16.91	1.33	8.24	83	300	52.02	77.61	13.74	0.81	6.45
37	600	40.48	51.78	8.99	0.65	6.14	83	576	8.31	14.64	3.44	0.16	3.36
37	900	4.91	9.11	2.4	0.09	2.49	83	600	7.51	10.65	2.54	0.12	3.19
37	971	0.47	1.58	0.64	0	1.36	83	682	0.25	0.71	0.45	0	1.01
37	1159	0	0	0	0	0.01	83	769	0	0.01	0	0	0.01
37	1133	0	0.01	0	0	0.01	83	900	0	0	0	0	0
37	1330	0	0	0	0	0	83	1240	0	0	0	0	0
37	1500	0	0	0	0	0	83	1510	0	0	0	0	0
37	1800	0	0	0	0	0	83	1809	0	0	0	0	0
37	1848	0	0	0	0	0	83	2114	0	0	0	0	0
							83	2500	0	0	0	0	0
							83	2800	0	0	0	0	0
							83	2999	0	0	0	0	0
Worst Run – Successful Run							Worst Run – Unsuccessful Run						
Run	G	ObjValue	Mean	Stddev	Variance	Pm	Run	G	ObjValue	Mean	Stddev	Variance	Pm
95	0	433.86	551.58	55.59	8.64	19.29	48	0	394.12	557.07	61.43	8.54	19.83
95	100	189.82	248.56	29.82	2.72	11.67	48	100	160.48	209.33	23.53	1.77	10.1
95	230	116.94	159.46	22.07	1.61	8.66	48	200	83.03	132.93	20.39	1.15	7.22
95	310	104.86	134.92	19.56	1.38	8.24	48	300	60.94	89.86	13.03	0.7	6.11
95	600	47	67.92	11.5	0.83	6.56	48	576	15.77	16.51	0.57	0	1.04
95	900	20.82	32.76	7.02	0.45	4.74	48	600	15.53	15.67	0.19	0	0
95	971	14.81	24.89	5.1	0.35	4.37	48	682	15.37	15.37	0	0	0.01
95	1159	0.69	1.85	0.62	0	1.04	48	769	15.36	15.36	0	0	0
95	1133	2.55	4.12	1.21	0.03	1.71	48	900	15.36	15.36	0	0	0
95	1330	0	0.01	0	0	0.01	48	1240	15.36	15.36	0	0	0
95	1500	0	0	0	0	0	48	1510	15.36	15.36	0	0	0
95	1800	0	0	0	0	0	48	1809	15.36	15.36	0	0	0
95	2083	0	0	0	0	0	48	2114	15.36	15.36	0	0	0
							48	2500	15.36	15.36	0	0	0
							48	2800	15.36	15.36	0	0	0
							48	2999	15.36	15.36	0	0	0

Based on the overall results the most competitive variants were $rand/1/bin$, $rand/2/bin$, $best/2/bin$ and $rand-to-best/1/bin$, they all could provide global optimum in all the function classes. On the other hand the worst results were provided by the variants $best/1/bin$, $best/1/exp$, $current-to-best/1/exp$ and $current-to- $rand/1/exp$$. The variant $best/2/exp$ could provide good result only for the function $f03$. Finally, the

other five variants *rand/1/exp*, *rand/2/exp*, *rand-to-best/1/exp*, *current-to-best/1/bin* and *current-to-rand/1/bin* were continuously shown different performances.

5 Conclusion

In this paper, we presented various empirical measurements on convergence property of DE variants. Comparison of fourteen DE variants to solve fourteen global optimization problems is done. Regardless of the characteristics and dimension of the functions, relatively better results seem to have been provided by the variants with binomial crossover. The reasons for poor performance of the other variants are identified as due to either stagnation or premature convergence, which are lead by improper balance between exploration and exploitation processes, this is evident in our results. This work can be still further analyzed by focusing on improving the performance of variants in the light of bringing balance between exploration and exploitation during the generations.

References

1. Storn, R., Price, K.: Differential Evolution – A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces. Technical Report TR-95-012, ICSI (March 1995)
2. Storn, R., Price, K.: Differential Evolution – A Simple and Efficient Heuristic Strategy for Global Optimization and Continuous Spaces. *Journal of Global Optimization* 11, 341–359 (1997)
3. Price, K.V.: An Introduction to Differential Evolution. In: Corne, D., Dorigo, M., Glover, F. (eds.) *New Ideas in Optimization*, pp. 79–108. Mc Graw-Hill, UK (1999)
4. Price, K., Storn, R.M., Lampinen, J.A.: *Differential Evolution: A practical Approach to Global Optimization*. Springer, Heidelberg (2005)
5. Vesterstrom, J., Thomsen, R.: A Comparative Study of Differential Evolution Particle Swarm Optimization and Evolutionary Algorithm on Numerical Benchmark Problems. In: *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2004)*, vol. 3, pp. 1980–1987 (2004)
6. Mezura-Montes, E., Velazquez-Reyes, J., Coello Coello, C.A.: A Comparative Study on Differential Evolution Variants for Global Optimization. In: *GECCO 2006* (2006)
7. Zaharie, D.: On the Explorative Power of Differential Evolution Algorithms. In: *3rd Int. Workshop Symbolic and Numeric Algorithms of Scientific Computing-SYNASC 2001*, Romania (2001)
8. Zaharie, D.: Critical values for the control parameters of Differential Evolution algorithms. In: *Proc. of the 8th International Conference of Soft Computing*, pp. 62–67 (2002)
9. Zaharie, D.: Control of Population Diversity and Adaptation in Differential Evolution Algorithms. In: Matouek, R., Omera, P. (eds.) *Proceedings of Mendel, Ninth International Conference on Soft Computing*, pp. 41–46 (2003)
10. Beyer, H.-G.: On the Explorative Power of ES/EP-like Algorithms. In: Porto, V.W., Waagen, D. (eds.) *EP 1998*. LNCS, vol. 1447, pp. 323–334. Springer, Heidelberg (1998)
11. Yao, H., Liu, Y., Lian, K.H., Lin, G.: Fast Evolutionary Algorithms. In: Rozenberg, G., Back, T., Eiben, A. (eds.) *Advances in Evolutionary Computing Theory and Applications*, pp. 45–94. Springer, New York (2003)
12. Personal Communication.: Unpublished
13. Feoktistov, V.: *Differential Evolution In Search of Solutions*. Springer, Heidelberg (2006)

An Intelligent System for Web Usage Data Preprocessing

V.V.R. Maheswara Rao¹, V. Valli Kumari², and K.V.S.V.N. Raju²

¹ Professor, Department of Computer Applications,
Shri Vishnu Engineering College for Women, Bhimavaram, W.G. Dt, Andhra Pradesh, India
mahesh_vvr@yahoo.com

² Professor, Department of Computer Science & Systems Engineering,
College of Engineering, Andhra University, Visakhapatnam, Andhra Pradesh, India
vallikumari@gmail.com, kvsvn.raju@gmail.com

Abstract. Web mining is an application of data mining technologies for huge data repositories. Before applying web mining techniques, the data in the web log has to be pre-processed, integrated and transformed. As the World Wide Web is continuously and rapidly growing, it is necessary for the web miners to utilize intelligent tools in order to find, extract, filter and evaluate the desired information. The data preprocessing stage is the most important phase in the process of web mining and is critical and complex in successful extraction of useful data. The web log is incremental in nature, thus conventional data preprocessing techniques were proved to be not suitable as they assume that the data is static. The web logs are non scalable, impractical and are distributed in nature. Hence we require a comprehensive learning algorithm in order to get the desired information.

This paper introduces an intelligent system, capable of preprocessing web logs efficiently. It can identify human user and web search engine accesses intelligently, in less time. The system discussed reduces the error rate and improves significant learning performance of the learning algorithm. The work ensures the goodness of split by using popular measures like Entropy and Gini index. The experimental results proving this claim are given in this paper.

Keywords: ISWUP, Human user accesses, Search engine accesses, session identification.

1 Introduction

Web mining is an application of data mining technology for huge web data repositories. The web mining can be used to discover hidden patterns and relationships with in the web data. The web mining task can be divided into three general categories, known as Web Content Mining, Web Structure Mining and Web Usage Mining.

The general process of web mining includes (i) Resource collection: Process of extracting the task relevant data, (ii) Information pre processing: Process of cleaning, Integrating and Transforming of the result of resource collection, (iii) Pattern discovery: Process of uncovered general patterns in the pre process data and (iv) Pattern analysis: Process of validating the discovered patterns.

(i)Resource collection: The conventional data mining techniques assumes that the data is static, and is retrieved from the conventional databases. In web mining techniques the nature of the data is incremental and is rapidly growing. One has to collect the data from web which normally includes web content, web structure and web usage. Web content resource is collected from published data on internet in several forms like unstructured plain text, semi structured HTML pages and structured XML documents.

(ii)Information pre processing: In conventional data mining techniques information pre processing includes data cleaning, integration, transformation and reduction. In web mining techniques the information pre processing includes a) Content pre processing, b) Structure pre processing and c) Usage pre processing. Content Preprocessing: Content preprocessing is the process of converting text, image, scripts and other files into the forms that can be used by the usage mining. Structure Preprocessing: The structure of a website is formed by the hyperlinks between page views. The structure preprocessing can be treated similar to the content pre processing. Usage Preprocessing: The inputs of the preprocessing phase may include the web server logs, referral logs, registration files, index server logs, and optional usage statistics from a previous analysis. The outputs are the user session files, transaction files, site topologies and page classifications.

iii)Pattern discovery: All the data mining techniques can be applied on preprocessed data. Statistical methods are used to mine the relevant knowledge.

iv)Pattern analysis: The goal of pattern analysis is to eliminate the irrelative rules and to extract the interesting rules from the output of patterns discovery process.

As the World Wide Web is continuously and rapidly growing, it is necessary for users to utilize intelligent tools in order to find, extract, filter, and evaluate the desired information and resources.

This paper introduces an Intelligent System Web Usage Preprocessor (ISWUP), which works based on a learning algorithm. The main idea behind ISWUP is to separate the human user accesses and web search engine accesses of web log data. This ISWUP acquires the knowledge from the derived characteristics of web sessions. It discards the web search engine accesses from the web log.

This paper is organized as follows. In section 2, we described related work. In next section 3, we introduced the overview of proposed work. In subsequent section 4, we expressed the study of theoretical analysis. In section 5, the experimental analysis of proposed work is shown. Finally in section 6 we mention the conclusions.

2 Related Work

Many of the previous authors are expressing the importance, criticality and efficiency of *data preparation* stage in the process of web mining. Most of the works in the literature do not concentrate on data preparation.

Myra Spiliopoulou [1] suggests applying Web usage mining to website evaluation to determine needed modifications, primarily to the site's design of page content and link structure between pages. Such evaluation is one of the earliest steps, that adaptive sites automatically change their organization and presentation according to the preferences of the user accessing them. M. Eirinaki and M. Vazirgiannis.[2] proposed a

model on web usage mining activities of an on-going project, called Click World, that aims at extracting models of the navigational behavior of users for the purpose of website personalization. However, these algorithms have the limitations that they can only discover the simple path traversal pattern, i.e., a page cannot repeat in the pattern. To extract useful web information one has to follow an approach of collecting data from all possible server logs which are non scalable and impractical. Hence to perform the above there is a need of an intelligent system which can integrate, pre process all server logs and discard unwanted data. The output generated by the intelligent system will improve the efficiency of web mining techniques with respect to computational time.

3 Overview of the Proposed Work

The web usage mining is a task of applying data mining techniques to extract useful patterns from web access logs. These patterns discover interesting characteristics of site visitors. Generally the web access logs are incremental, distributed and rapidly growing in nature. It is necessary for web miners to utilize intelligent tools / heuristic functions in order to find, extract, filter and evaluate the desired information.

Before applying web mining techniques to web usage data, the web usage resource collection has to be cleansed, integrated and transformed. To perform the same first it is important to separate accesses made by human users and web search engines. Web search engine is a software program that can automatically retrieve information from the web pages. Generally these programs are deployed by web portals. To analyze user browsing behavior one must discard the accesses made by web search engines from web access logs. After discarding the search engine accesses from web access logs, the remaining data are considered as human accesses.

3.1 Intelligent Systems

The intelligent system takes the raw web log as input and discards the search engine accesses automatically with less time. It generates desired web log consists of only human user accesses. The web log preprocessing architecture shown in Figure 1.

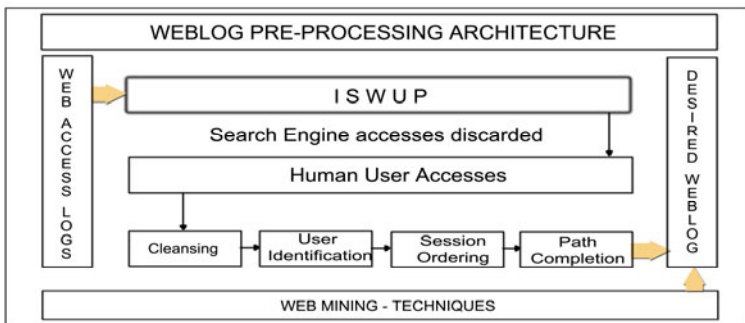


Fig. 1. Web log preprocessing architecture

3.1.1 Types of Intelligent Systems

The intelligent systems can be broadly categorized into server side intelligent systems and client side intelligent systems. The client side heuristic functions are further categorized into resource collection and information pre processing heuristic functions. The present paper introduces a learning heuristic function to separate human accesses and web search engine accesses from web access logs as a primary step of pre processing of web log data.

3.1.2 Working of ISWUP

The main goal of ISWUP is to separate the human user and search engines accesses. To perform this task any intelligent system requires a learning capability. Any intelligent system acquires the knowledge from the knowledge base, where knowledge base is a “set of related facts”. All the records in the web access logs are taken as testing data. *Derived attributes* from web logs can be considered as characteristics, which separate human user and web search engine accesses.

Total pages, inner pages, total time spent, repeated access, get, post and so on are called derived attributes. The derived attributes can be taken as a set of facts and form a knowledge base. This knowledge base can be used as training data to the ISWUP. The web usage pre processing includes cleansing, user identification, session identification, path completion and formatting. The raw web usage data collected from different resources in web access log includes IP address, unique users, requests, time stamp, protocol, total bytes and so on as shown in Table 1.

Table 1. Sample web log

S.No	IP Address	Unique Users	Requests	Time Stamp	Protocol	Total Bytes

To label the web sessions the ISWUP takes the training data as characteristics of session identification. A web session is a sequence of request made by the human user or web search made during a single visit to a website. This paper introduces a learning tree known as ISWUP to accomplish above task.

The ISWUP learning tree consists of root node, internal node and leaf of terminal node. A root node that has no incoming edges and two or more outgoing edges. Any internal node has exactly one incoming edge and two or more outgoing edges. The leaf or terminal node each of which has exactly one incoming edge and no outgoing edges. In ISWUP learning tree, each leaf node is assigned with a class label. The class labels are human user access session and web search engine access sessions. The root node and other internal nodes are assigned with the characteristics of the session. The ISWUP learning tree works on a repeatedly posing series of questions about the characteristics of the session identification and it finally yields the class labels.

3.1.3 Modeling of ISWUP

The ISWUP learning tree can be constructed from a set of derived attributes from knowledge base. An efficient ISWUP learning tree algorithm has been developed to get reasonably accurate learning to discard web search engine accesses from web log accesses. The algorithm is developed based on the characteristics of session.

Based on the tree traversal there are two notable features namely depth and breadth. Depth determines the maximum distance of a requested page where distance is measured in terms of number of hyperlinks from the home page of website. The breadth attribute determines the possible outcomes of each session characteristics. The proposed model suggests the following characteristics to distinguish human user accesses and web search engine accesses.

- ❖ Accesses by web search engine tend to be more broad where as human accesses to be of more depth.
- ❖ Accesses by web search engines rarely contain the image pages whereas human user accesses contain all type of web pages.
- ❖ Accesses by web search engines contain large number of requested pages where as human user accesses contain less number of requested pages.
- ❖ Accesses by the web search engines are more likely to make repeated requests for the same web page, where as human users accesses often make repeated requests.

3.1.4 Algorithm for Intelligent System ISWUP

TreeExtend(DA, TA)

- 1: If ConditionStop(DA, TA) = True then
- 2: TerminalNode = CreateNewNode()
- 3: TerminalNode.Label = AssignedLabel(DA)
- 4: Return TerminalNode
- 5: Else
- 6: Root = CreateNewNode()
- 7: Root.ConditionTest = DeriveBestSplit(DA, TA)
- 8: Let $V = \{v / v \text{ is a possible outcome of ConditionTest}()\}$
- 9: For each $v \in V$ do
- 10: $DA_v = \{da / \text{Root.ConditionTest}(da) = v \text{ and } d \in DA\}$
- 11: Child = TreeExtend(DA_v , TA)
- 12: Add Child as descendant of root and label the edge as v
- 13: End for
- 14: End if
- 15: Return root

The input to the above algorithm consists of Training data DA and Testing data TA. The algorithm works by recursively selecting DeriveBestSplit() (step 7) and expanding the leaf nodes of the tree (Step 11 & 12) until condition stop is met (Step1). The details of methods of algorithm are as follows

CreateNewNode(): This function is used to extend the tree by creating a new node. A new node in this tree is assigned either a test condition or a class label.

ConditionTest(): Each recursive step of TreeExtend must select an attribute test condition to divide into two subsets namely human user accesses and search engine accesses. To implement this step, algorithm uses a method ConditionTest for measuring goodness of each condition.

ConditionStop(DA, TA): This function is used to terminate the tree extension process by testing whether all the records have either the same class label or the same

attribute values. Another way of stopping the function is to test whether the number of records have fallen below minimum value.

AssignLabel (): This function is used to determine the class label to be assigned to a terminal node. For each terminal node t , Let $p(i/t)$ denotes the rate of training records from class i associated with the node t . In most of the cases the terminal node is assigned to the class that has more number of training records.

DeriveBestSplit () : This function is used to determine which attribute should be selected as a test condition for splitting the training records. To ensure the goodness of split, the Entropy and Gini index are used.

3.1.5 Example for ISWUP

The main idea of ISWUP is to label the human user accesses and search engine accesses separately. The intelligent system acquires the knowledge from the derived characteristics of web log as shown in Table 3. Using ISWUP algorithm the derived characteristics are assigned to root node and intermediate nodes of the tree as shown in figure 2. The leaf nodes are labeled with human user or search engine accesses.

Table 2. Example of web server log

No	IP Address	Unique Users	Requests	Time Stamp	Protocol	Total Bytes
1	125.252.226.42	1	4	11/22/2009 12:30	HTTP\1.1	14.78 MB
2	64.4.31.252	1	69	11/22/2009 13:00	HTTP\1.1	782.33 KB
3	125.252.226.81	1	41	11/22/2009 13:30	HTTP\1.1	546.71 KB
4	125.252.226.83	1	19	11/22/2009 14:00	HTTP\1.1	385.98 KB
5	125.252.226.80	1	20	11/22/2009 14:30	HTTP\1.1	143.44 KB
6	58.227.193.190	1	18	11/22/2009 15:00	HTTP\1.1	108.99 KB
7	70.37.129.174	1	4	11/22/2009 15:30	HTTP\1.1	86.66 KB
8	64.4.11.252	1	2	11/22/2009 16:00	HTTP\1.1	52.81 KB
9	208.92.236.184	1	17	11/22/2009 16:30	HTTP\1.1	32.13 KB
10	4.71.251.74	1	2	11/22/2009 17:00	HTTP\1.1	25.82 KB

Table 3. Example of Characteristics / Derived Attributes

Derived Attribute	Description
Total pages	Total pages retrieved in a web session
Image pages	Total number of image pages retrieved in a web session
Total Time	Total amount of time spent by website visitor
Repeated access	The same page requested more than once in a web session
Error request	Errors in requesting for web pages
GET	Percentage of requests made using GET method
Breadth	Breadth of the web traversal
Depth	Depth of the web traversal
Multi IP	Session with multiple IP addresses

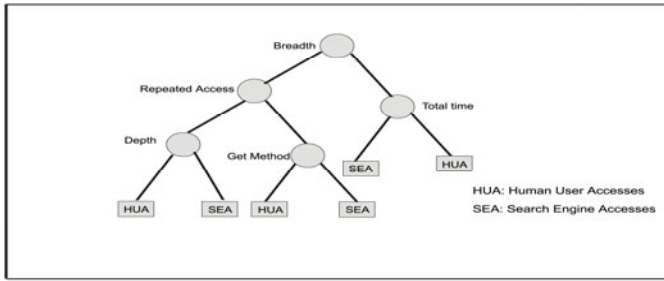


Fig. 2. Example of Learning Tree

3.2 Path Completion

Path completion is process of adding the page accesses that are not in the web log but those which be actually occurred. A mechanism is developed to infer the missing pages and generate the time stamp for missing pages.

4 Theoretical Analysis

The learning performance of any algorithm is proportionate on the training of algorithm, which directly depends on the training data. As testing data is continuously growing the training data is also continuous. Hence to estimate the training data one can use predictive modeling technique called regression. The goal of regression is to estimate the testing data with minimum errors.

Let S denote a data set that contains N observations,

$$S = \{(D_i, T_i) / i = 1, 2, 3, \dots, N\}$$

Suppose to fit the observed data into a linear regression model, the line of regression D on T is

$$D = a + bT \tag{1}$$

Where a and b are parameters of the linear model and are called regression coefficients. A standard approach for doing this is to apply the method of least squares, which attempts to find the parameters (a, b) that minimize the sum of squared error say E.

$$E = \sum_{i=1}^n (D_i - a - bT_i)^2 \tag{2}$$

The optimization problem can be solved by taking partial derivative of E w.r.t a and b, equating them to zero and solving the corresponding system of linear equations.

$$\frac{\partial E}{\partial a} = 0 \quad \Rightarrow \quad \sum_{i=1}^n D_i = na + b \sum_{i=1}^n t_i \tag{3}$$

$$\frac{\partial E}{\partial b} = 0 \quad \Rightarrow \quad \sum_{i=1}^n D_i t_i = a \sum_{i=1}^n t_i + b \sum_{i=1}^n t_i^2 \tag{4}$$

Equations (3) and (4) are called normal equations. By solving equations (3) and (4) for a given set of D_i, T_i values, we can find the values of ‘a’ and ‘b’, which will be the best fit for the linear regression model. By dividing equation (3) by ‘N’ we get

$$\bar{D} = a + b\bar{T} \tag{5}$$

Thus the line of regression D on T passes through the point (\bar{D}, \bar{T})

We can define, $\mu_{11} = Cov(D, T) = \frac{1}{n} \sum_{i=1}^n D_i T_i - \bar{D}\bar{T}$

$$\Rightarrow \frac{1}{n} \sum_{i=1}^n D_i T_i = \mu_{11} + \bar{D}\bar{T} \tag{6}$$

Also $\frac{1}{n} \sum D_i^2 = \sigma_d^2 + \bar{D}^2$ (7)

From equations (4), (6) and (7) we get

$$\mu_{11} + \bar{D}\bar{T} = a\bar{D} + b(\sigma_d^2 + \bar{D}^2) \tag{8}$$

And on simplifying (8), we get

$$\mu_{11} = b\sigma_d^2 \Rightarrow b = \frac{\mu_{11}}{\sigma_d^2} \tag{9}$$

b is called the slope of regression D on T and the regression line passes through the point (\bar{D}, \bar{T}) . The equation of the regression line is

$$D - \bar{D} = b(T - \bar{T}) = \frac{\mu_{11}}{\sigma_d^2} (T - \bar{T})$$

$$D - \bar{D} = r \frac{\sigma_d}{\sigma_t} (T - \bar{T})$$

$$\Rightarrow D = \bar{D} + r \frac{\sigma_d}{\sigma_t} (T - \bar{T}) \tag{10}$$

The linear regression coefficient ‘r’ is used to predict the error between testing data and training data. The learning performance can also be expressed in terms of training error rate of the learning algorithm. The training error rate is given by the following equation,

$$\text{Training Error Rate} = \frac{\text{Number of wrong characteristic definitions}}{\text{Total number of characteristic definitions}}$$

5 Experimental Study

The experiments are conducted on one day web log data. The results are analyzed and are shown in Figure 3. The error rate between the testing data and training data is almost minimized and is found to be 0.2 on the average. Hence the experimental study is in line with the theoretical analysis of, goal of regression. The nature of relation between testing data and training data is studied and both are proven as continuous. The execution performance of the algorithm is constantly improving up to certain training data, from that it improves drastically as indicated in Figure 4.

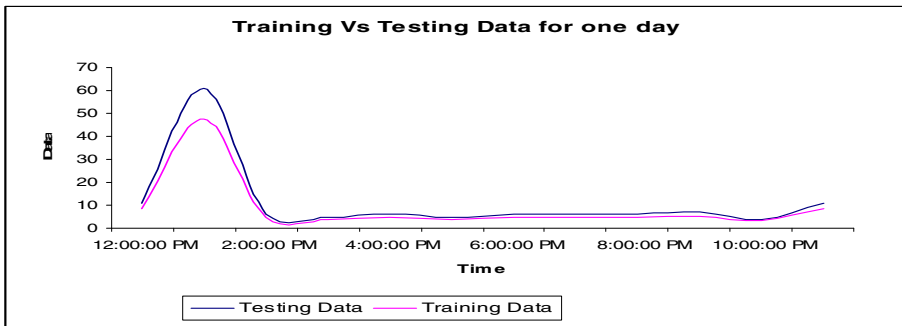


Fig. 3. Training Vs Testing Data

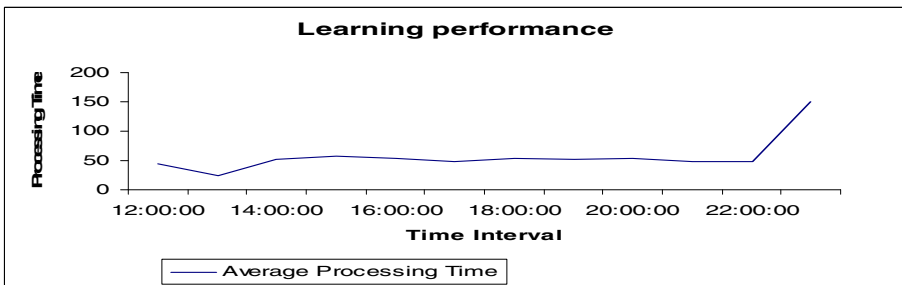


Fig. 4. Learning performance

6 Conclusion

Web mining is a new and promising research issue to help users in gaining insight into overwhelming information on the World Wide Web. The present paper discusses about the importance and criticality of web log pre-processing, including the definition, architecture and all stages of it. And the present paper introduces an intelligent

system, which improves the efficiency of preprocessing of web logs. It separates human user and web search engine accesses automatically, in less time. It reduces the error rate learning algorithm. It ensures the goodness of split by taking widely using measures like Entropy and Gini index.

References

1. Spiliopoulou, M.: Web Usage Mining for Site Evaluation. *Comm. ACM* 43(8), 127–134 (2000)
2. Eirinaki, M., Vazirgiannis, M.: Web mining for web personalization. *ACM Transactions on Internet Technology (TOIT)* 3(1), 1–27 (2003)
3. Kleinberg, J.M.: Authoritative sources in a hyperlinked environment. In: *ACM-SIAM Symposium on Discrete Algorithms* (1998)
4. Kamdar, T.: *Creating Adaptive Web Servers Using Incremental Weblog Mining*, masters thesis, Computer Science Dept., Univ. of Maryland, Baltimore, CO–1 (2001)
5. Wang, Y.: *Web Mining and Knowledge Discovery of Usage Patterns* (February 2000)
6. Cooley, R., Mobasher, B., Srivastava, J.: Web mining: Information and pattern discovery on the World Wide Web. In: *Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 1997* (1997)
7. Srivastava, J., Desikan, P., Kumar, V.: Web Mining: Accomplishments and Future Directions. In: *Proc. US Nat'l Science Foundation Workshop on Next-Generation Data Mining (NGDM)*. Nat'l Science Foundation (2002)
8. Kumar, R., et al.: Trawling the Web for Emerging Cybercommunities. In: *Proc. 8th World Wide Web Conf.*, Elsevier Science, Amsterdam (1999)
9. Manolopoulos, Y., et al.: *Indexing Techniques for Web Access Logs*. Web Information Systems, IDEA Group (2004)
10. Armstrong, R., et al.: Webwatcher: A Learning Apprentice for the World Wide Web. In: *Proc. AAAI Spring Symp. Information Gathering from Heterogeneous, Distributed Environments*. AAAI Press, Menlo Park (1995)
11. Chen, M.-S., Park, J.S., Yu, P.S.: Efficient Data Mining for Path Traversal Patterns. *IEEE Trans. Knowledge and Data Eng.* 10(2) (1998)
12. Yanchun, C.: Research on Intelligence Collecting System[J]. *Journal of Shijiazhuang Railway Institute(Natural Science)* (2008)
13. *Proceedings of the IEEE International Conference on Tools with Artificial Intelligence* (1999)
14. Chen, M.S., Park, J.S., Yu, P.S.: Efficient Data Mining for Path Traversal Patterns in a Web Environment. *IEEE Transaction on Knowledge and Data Engineering* (1998)

Discovery and High Availability of Services in Auto-load Balanced Clusters

Shakti Mishra¹, Alok Mathur¹, Harit Agarwal¹,
Rohit Vashishtha², D.S. Kushwaha¹, and A.K. Misra¹

¹ Computer Science & Engineering Department
Motilal Nehru National Institute of Technology, Allahabad, India
{shaktimishra, el075033, it078010, dsk, akm}@mnnit.ac.in

² Member Technical Staff, Oracle, India
rohit7mnnit2k7@gmail.com

Abstract. This paper presents an approach for service discovery and service sharing in an auto-load balanced cluster environment. We propose a B+ tree based architectural framework for dynamic configuration of cluster to achieve load balancing. Our prime focus is on the discovery and sharing of services in the cluster in auto-configurable load balanced cluster. The load balancing is required because there are instances when a node is burdened with too many connections accessing a particular service. This demands replication of service to an under loaded node so as to optimize the cluster performance. The implementation of proposed approach is carried out by using MOSIX as a middleware. Experimental results and simulations shows that the configuration cost of a cluster size of 50-60 nodes have lowest execution time of 2200 microseconds, which is significantly better than other approaches. Our mechanism is able to show that replication of service on the idle nodes of local or same cluster keeps the rate of traffic significantly low, which in turn increases the overall performance of the cluster.

Keywords: Cluster, Service, High Availability, Load balancing, MOSIX.

1 Introduction

A distributed system consists of multiple autonomous computers that can be classified on the basis of number of nodes, location of machines and resource sharing. Major architectures are cluster, grid, peer to peer, cloud computing etc. [1].

Clustering enables a group of independent nodes to be managed as a single system for higher availability; easier manageability and greater scalability. In clusters or network of workstations (NOW), more than two nodes are interconnected via communication link [10]. The proximity of nodes may be on the same location or at different physical location giving a single coherent view of the whole system to user. These machines may be single or multiprocessor system with memory and resources (I/O, Network bandwidth etc.). With clustering, there is a fairly tight integration between the nodes in the cluster. However, when failure occurs in a cluster, resources are redirected and the workload is redistributed [2].

Discovery of services in cluster environment can be done in two ways; the first option is to maintain a central directory of services. However, the major problem associated with this approach is the huge size of this directory. The second option is to use decentralized directory, which generates huge network traffic (in form of messages). Third option is to build an overlay network which contains the best part of first and second option. Services are accessed in random fashion and hence there is always imbalance in the numbers of consumers and providers for any particular service. A service owned by small numbers of nodes might be required by a huge number of users. If this happens, then these service provider nodes become overloaded due to large numbers of requests. In this paper, we also propose a solution to avoid this problem, and propose an architecture in which any system may allow to use its service by other systems without any load imbalance.

Service Oriented Architecture (SOA) is a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains and implemented using various technology stacks. One perceived value of SOA is that it provides a powerful framework for matching needs and capabilities and for combining capabilities to address those needs. One usage may be repurposed across a multitude of needs [8, 9]. SOA is a “view” of architecture that focuses in on services as the action boundaries between the needs and capabilities in a manner conducive to service discovery and repurposing. It is an architectural paradigm and discipline that may be used to build infrastructures enabling those needs (consumers) and those with capabilities (providers) to interact via services across disparate domains of technology and ownership.

A distributed service discovery architecture that relies on a virtual backbone for locating and registering available services was presented in [3]. The disadvantage of this approach is that it totally relies on multicasting and broadcasting techniques for service discovery and registration. The distributed collaborative service discovery and service sharing framework [4] defines three types of nodes: service directories, service providers and requesting nodes. Service directory nodes act as mediators for lookup requests from requesting nodes. The basic limitation was with this approach is to forward the request message sequentially, which in turn increases network communication overhead.

To address some of these issues, we present an approach for modeling the dynamic behavior of clusters for load balancing. Load balancing in cluster depends upon the joining and leaving of nodes. We propose a dynamic configuration of clusters for load balancing that also incorporates a fault resilient mechanism to handle the crash and failure of nodes. To run our application in the proposed architecture, we propose discovery of services.

2 Dynamic Cluster Configuration: Structure and Approach

We consider a cluster network (Fig. 1) in which nodes are loosely connected to each other via intercommunication network. Our proposal is based on a B^+ tree to emulate dynamic configuration of cluster in which each node of the tree is maintained by a

computing node. The nodes at leaf level are assumed to be connected with each other via communication link. Fig. 2 presents the proposed B^+ tree based hierarchical structure of clusters. A group of clusters is monitored by an Inter Cluster head (ICH) node, which are further administrated by Process Management Server (PMS) nodes [11]. All PMS and ICH nodes maintains direct communication link with their adjacent nodes to exchange state information of clusters. The load computing criteria for each node is taken as CPU utilization. A threshold value for load is identified for under loaded and over loaded nodes.

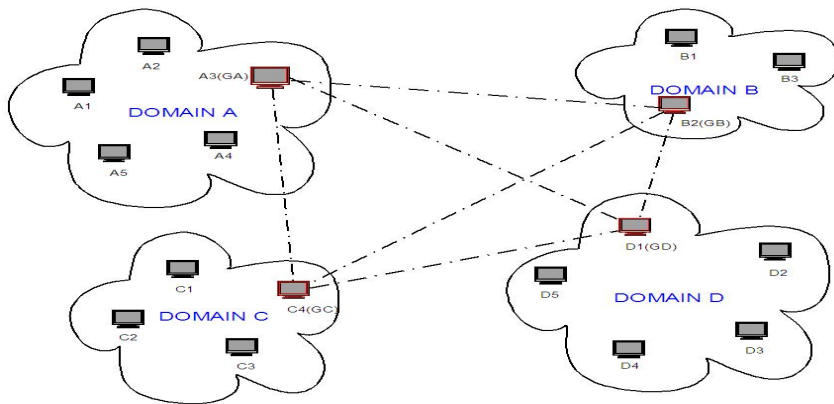


Fig. 1. Basic Architecture of Proposed Infrastructure

The nodes in a cluster are organized in an order of increasing load so that the left most node of each cluster is the least loaded one that also monitors the rest of the node. The load index is taken as integer variable associated with each node. The number of leaf nodes in a cluster is selected on the basis of the network configuration and bandwidth. However, number of clusters under one ICH has a threshold limit. Each ICH node maintains pointers to the least loaded node of cluster among all the leaf nodes. The PMS node monitors the Inter cluster heads and receives heartbeat message from them. PMS nodes also replicate themselves to keep the backup of nodes and to provide information when an ICH fails.

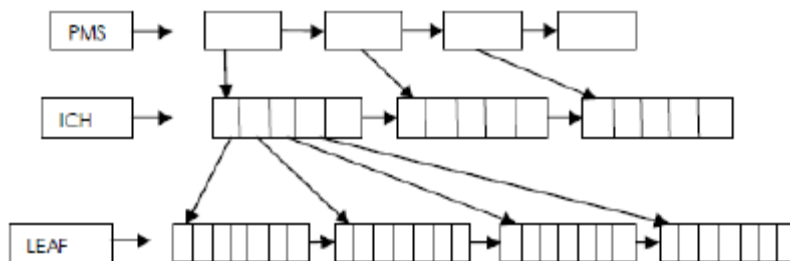


Fig. 2. B^+ tree mapping of proposed infrastructure

2.1 Data Structure Used

PMS Class

- ICH node pointer: This is the pointer to point to the corresponding ICH node.
- PMS node pointer: This is the pointer to point to the next PMS node.
- IP address: This presents the IP address of node.

ICH Class

- ICH value array: This array contains value of inter cluster head of the cluster.
- Leaf pointer array: This array contains the pointer of leaf corresponding to each ICH.
- ICH node pointer: This pointer points to the next ICH node.
- Number: This integer represents the number of elements currently in ICH.

LEAF Class

- Load array: This array contains the load of nodes.
- Load_type array: This array contains the type of loads (under load, average load, overload), corresponding to each load.
- Leaf node pointer: This contain pointer to point to the next Leaf node.
- Number: This integer represents the number of elements or load currently in leaf node.
- Count_load: This integer contains the number of overloaded elements in Leaf node.

2.2 Cluster Components Used

Local_Cluster_Members: Local_Cluster_Members list contains the IP addresses of the nodes of the assigned cluster.

Remote_group_leader_list: It contains the IP address of CH nodes of other clusters. The major entry includes the cluster name and its corresponding cluster head IP address.

Underloaded_local_node_list: This list consists of the IP address of the under-loaded nodes of local cluster.

Underloaded_Remote_node_list: Each Cluster head maintains this list. This list consists of IP address of under-loaded nodes of other clusters. When any node queries the cluster head for a underloaded node list of other clusters, then cluster head randomly selects some of the under loaded nodes IP and sends it to the querrying node.

Self_Service list: Each node maintains list of services that are offered by it. Each node exchanges this list with every other node of its local cluster.

Local_cluster_services list: It contains all the services available within local cluster and IP address service provider nodes. It can be formed by exchanging the self_service list within the local cluster.

3 Proposed Algorithms and Functions

3.1 Algorithms

Leaf/ Node Insertion

1. Scan the Load value to be inserted in List.
2. If the link list is empty, create the object of ICH and Leaf class and enter the value in Leaf and ICH else traverse the ICH list.
3. If an element to be entered is not overload, then search the leaf which is not full. If all leaf is full, insert element in first leaf and divide the leaf.
4. If element to be entered is overload, then search the leaf which is not overloaded. If all leaf are overloaded then reject the element.
5. When the leaf is selected, use the insertion sort to place the element in order.
6. If the leaf node in which element is to be inserted is full, then call Leaf Split.

Leaf Deletion

1. The only factor to be considered here is that when the element is found, we delete that element and if the element deleted was the only element in leaf, then we delete the leaf node and delete its pointer from ICH node and decrease the number of element in ICH node.
2. Call ICH Merge.

Leaf Merge

1. If the leaf contains the number of elements below threshold, then merge the elements of this leaf node with its adjacent node.
2. After merging, if the number of elements in a leaf reaches above threshold, call Leaf Split().

Leaf Split

1. If number of elements in a leaf reaches above threshold after node insertion, create an empty leaf node and place alternate load from overloaded leaf to this leaf so that the load is balanced.
2. Check if an ICH is full, call ICH Division().
3. Update the ICH.

ICH Merge

1. To merge ICH with the adjacent ICH (iff total number of elements in both ICH are less than the maximum number of pre-defined elements in ICH node), then copy all elements of next ICH into this ICH and delete the next ICH.
2. Delete the entry of corresponding ICH from PMS.

ICH Division

1. Create array of size (MAXIMUM no. of element in ICH +1) of ICH_divide structures.
2. Struct ICH_divide { int val; //store value of ICH element Leaf node *br // store the pointer corresponding to ICH element }
3. Copy the ICH to be divided and new element for ICH into the array of struct ICH_divide & sort the array of struct ICH_divide according to load values.
4. Create a new object of ICH class & copy alternate elements of array if ICH_divide into both old ICH and new ICH object.

3.2 Functions

Load information exchange with other cluster heads

Cluster head contains the below_loaded_local_list, which includes the IP address of all the below loaded nodes in its cluster. If load status changes, the corresponding node sends the status update message to the cluster head and cluster head updates it below_loaded_local_list. This list is further exchanged with other cluster heads in order to find out below loaded node, if their own cluster becomes overloaded. So at anytime, cluster head contains the list of below_loaded_local_list and external_below-loaded_list which contains all the below loaded nodes of local cluster and some of the below loaded nodes of other clusters respectively.

Services information exchange with other cluster heads

In local cluster each node sends timely its own_service list to all other nodes and in turn each node updates its list of local_cluster_service.

When any node requires a service that is not in Self_Service it then checks in local_cluster_service. If that service is not found in both these tables, then it queries its cluster head for this service. Cluster head forwards this query to other cluster heads which in turn reply to the query if the service is available with the IP address of the service providing node.

4 Implementation and Experimental Results

4.1 Implementation of Auto-load Balanced Clusters

To evaluate the cost of algorithms for building the hierarchical tree structure, we first implemented our approach in C++ using socket programming. The approach is tested for different number of nodes N from 10 to 100 in a cluster. For a cluster of size N , $100 \times N$ values are inserted in batches. For each set of values, we have calculated the execution time. To simulate different sequences of events (leaf insertion and deletion), we have taken different number of ICH nodes to manage varying number of clusters.

```

ich elements
1      4      3      2
leaf elements1 7      18      12      20      21      31      32      84      87
leaf elements4 8      14      33      34      35      36      86      88      98
leaf elements3 6      11      13      37      38      39      48      75      85
leaf elements2 4      5      7      15      41      42      43      80      98
*****menu*****
1.insert
2.delete
3.display
4.exit
enter your choice
1
enter a value
44
value inserted
*****menu*****
1.insert
2.delete
3.display
4.exit
enter your choice
3
ich elements
1      3      7
leaf elements1 10     28     31     44
leaf elements3 6      11     13     37     38     39     48     75     85
leaf elements7 12     21     32     84     87
ich elements
2      4
leaf elements2 4      5      7      15     41     42     43     80     98
leaf elements4 8      14     33     34     35     36     86     88     98

```

Fig. 3. Leaf Insertion and Merge

```

ich elements
1      3      12
leaf elements1 10     28     31     44
leaf elements3 6      11     13     37     38     39     40     75     85
leaf elements12
ich elements
8
leaf elements8 14     33     34     35     36     86     88     98
*****menu*****
1.insert
2.delete
3.display
4.exit
enter your choice
2
enter a value to delete
12

entered in ich merge
*****menu*****
1.insert
2.delete
3.display
4.exit
enter your choice
3
ich elements
1      3      8
leaf elements1 10     28     31     44
leaf elements3 6      11     13     37     38     39     40     75     85
leaf elements8 14     33     34     35     36     86     88     98

```

Fig. 4. Leaf Deletion and ICH Division

Fig. 3 and 4 show the snapshots for different operations. The cost of insertion and deletion operations (in terms of execution time) is illustrated in Fig. 5. We also observe that the configuration cost of a cluster consisting of 50-60 nodes have lowest execution time of 2200 microseconds, which is significantly better than other approaches.

4.2 Experimental Study for Discovery of Services in Auto-load Balanced Clusters

To evaluate the application performance of our proposed model, we deployed a sample Cluster using 8 Nodes with MOSIX [6] as a middleware with heterogeneous



Fig. 5. Execution Time graph for Insertion and Deletion operation

```
File Edit View Terminal Help
NODE1: $./a.out
***** Welcome to Computers Grid System *****
..... Initializing Grid ....
.....
my ip is 172.31.72.42
...Timed out..... No response from any group leader ....
.....initializing this node as group leader .....
.....
..... Group Leader IP is my own ip 172.31.72.42
.....
□
```

Fig. 6. Initial searching for Cluster heads

```
File Edit View Terminal Help
Node4: $./a.out
***** Welcome to Computers Grid System *****
..... Initializing Grid ....
.....
My IP is 172.31.77.47
.....GOT A RESPONSE FROM GROUP LEADR .....
.....
The ip address of the Group Leader is : 172.31.77.41
In This Thread I will Reply for my Own Group Leader IP
□
```

Fig. 7. Initial searching for Cluster heads and Reply from cluster heads

configuration. The proposed algorithms are implemented in C using socket programming and shell scripting. The clusters are divided into four domains. We observe behavior of all these nodes at different time with different load conditions and following are the results with time and load variation. Initially, node 1, IP address 172.31/72.42 finds out cluster heads of its cluster as shown in Fig.6. It wait for some time and after it set itself as a cluster head as shown in Fig 7.

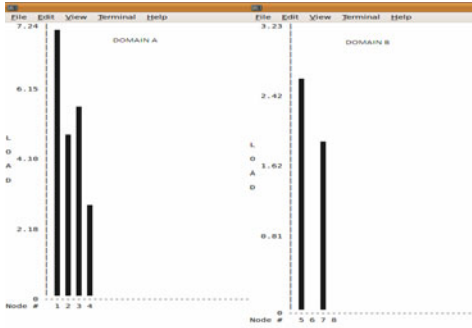


Fig. 8. Load status of clusters before sharing the services

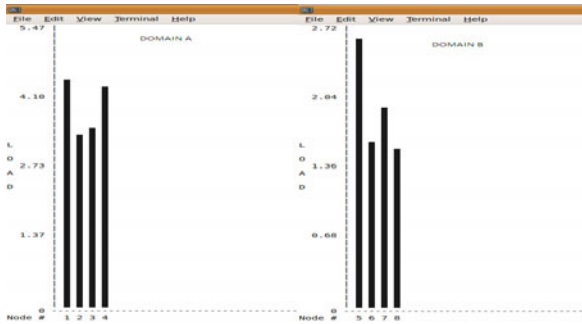


Fig. 9. After process migration: load status of cluster 1 and 2

After getting cluster heads get connected, by using separate threads, nodes exchange their service information about the services. Initially, node 6 and 8 have an empty other_service file and nodes in cluster 1 only contains self_service list. Fig.8 represents the load status of nodes in a cluster before sharing the services. But after some time, exchange of service information occurs and entries are updated in other_services file of cluster 2. We find that node 2 is a normal-loaded while node1 becomes over-loaded due to service sharing with node 1. Node 1 is not able to find out any below loaded node in its local cluster, so it finds below loaded node 6 in another cluster. Similarly, now, node1 replicates its service to node4 and by doing so the load status of each cluster becomes nearly equal (Fig. 9), which represents load balancing among all clusters.

5 Conclusion

We have established a decentralized self discovering B^+ tree based architecture where nodes are interconnected with each other so as to discover and share services of nodes while keeping the cluster load balance. Based on our proposal, discovery of services in this huge network of workstations (NOW) can also be carried out in a decentralized fashion. Sharing services with other nodes can increase the load on a particular node.

However, experimental results show that our proposed migration based mechanism of sharing services with idle nodes of local or same domain keeps the rate of traffic significantly lower, and increases the availability of services and thus overall performance of the cluster.

References

1. Liu, P.-W., Chan, L.-S., Ling, F.-Y., Hang, S.-S.: Distributed Computing: New Power for Scientific Visualization 16(3) (May 1996)
2. Buyya, R.: High Performance Cluster Computing. Pearson Education 1, 519–532 (2008)
3. Kozat, U., Tassiula, L.: Service discovery in mobile adhoc Networks: an Overall Perspective on architectural choices and network Layer support issues. *Adhoc Networks* 2(1), 23–44 (2004)
4. Safa, H., Artail, H., Hamze, H., Merhad, K.: A collaborative service discovery and service sharing framework for mobile ad hoc networks. In: Li, K., Jesshope, C., Jin, H., Gaudiot, J.-L. (eds.) *NPC 2007*. LNCS, vol. 4672, pp. 151–160. Springer, Heidelberg (2007)
5. Milojicic, D., HP Labs Douglass, F.: Process Migration Review. AT&T Labs-Research, Yves Paindaveine TOG research Institute, Richard Wheeler EMC and Songnain Zhou University of Toronto and Platform Computing published in *ACM Computing Surveys* 32(3), 241–299 (2000)
6. Barak, A., Shiloh, A.: The MOSIX Management System for Linux Clusters, Multi-Clusters and Clouds. White Paper
7. Caron, B., Garonne, V., Tsaregorodtsev, A.: Definition, modeling and simulation of a cluster computing scheduling system for high throughput computing. *Future Generation Computer Systems* 23, 968–976 (2007)
8. Service Oriented Architecture (SOA) and Specialized Messaging Patterns Technical white paper by Adobe
9. Yefim, V., Gartner, N.: Service-Oriented Architecture Scenario. publication ID Number: AV-19-6751 (2003)
10. Li, Y., Lan, Z.: A Survey of Load Balancing in Cluster Computing. Department of Computer Science, Illinois Institute of Technology, Chicago, IL 60616
11. Mishra, S., Kushwaha, D.S., Misra, A.K.: Hybrid Load Balancing in Auto-Configurable Load Balanced Cluster. *Journal of Computer Science & Engineering* 2(1), 16–25 (2010)

Bayes Theorem and Information Gain Based Feature Selection for Maximizing the Performance of Classifiers

Subramanian Appavu, Ramasamy Rajaram,
M. Nagammai, N. Priyanga, and S. Priyanka

¹ Faculty of Information Technology and Computer Science

² Pre-Final Year Students of Information Technology,
Thiagarajar College of Engineering, India

Abstract. Features play a very important role in the task of pattern classification. Consequently, the selection of suitable features is necessary as most of the raw data might be redundant or irrelevant to the recognition of patterns. In some cases, the classifier can not perform well because of the large number of redundant features. This paper presents a novel evolving feature selection algorithms taking the advantages of Bayes Theorem and Information Gain to improve the predictive accuracy. Bayes theorem is used to discover dependency information among features. In addition to that, feature selection has been improvised by Information Gain which selects features based on their importance. Different features play different roles in classifying datasets. Unwanted features will result in error information during classification which will reduce classification precision. The proposed feature selection can remove these distractions to improve classification performance. As shown in the experimental results, after feature selection using the Bayes theorem and Information gain to control false discovery rate, the classification performance of DT's and NB classifiers were significantly improved.

Keywords: Data mining, Feature Selection, Classification, Bayes Theorem and Information Gain.

1 Introduction

Feature selection is one of the important and frequently used techniques in data pre-processing for data mining. It reduces the number of features, removes irrelevant, redundant, or noisy data, and brings the immediate effects for applications, speeding up a data mining algorithms and improving the mining performance. Selecting the right set of features for classification is one of the most important problems in designing a good classifier [7]. Very often we don't know a-priori what the relevant features are for a particular classification tasks. One popular approach to address this issue is to collect as many features as we can prior to the learning and data-modelling phase. However, irrelevant or correlated features, if present, may degrade the performance of the classifier.

In the emerging area of data mining applications, users of data mining tools are faced with the problem of datasets that are comprised of large number of features and instances. Such kinds of datasets are not easy to handle for mining. The mining process can be made easier to perform by focussing on a set of relevant features while ignoring the other ones.

In this paper, we present our study on features subset selection and classification with the DT's and NB algorithm. In Section 2, we briefly describe the survey on feature selection methods. The Problem statement is given in Section 3. The Proposed method is described in Section 4. In Section 5, we describe our results. The conclusion is given in Section 6.

2 The Literature Review

In general, feature selection techniques can be split into two categories - filter methods [3] and wrapper methods [8]. Wrapper methods generally result in better performance than filter methods because the feature selection process is optimized for the classification algorithm to be used. However, they are generally far too expensive to be used if the number of features is large because each feature set considered must be evaluated with the trained classifier. Filter methods are much faster than wrapper methods and therefore are better suited to high dimensional data sets. Diverse feature ranking and feature selection techniques have been proposed in the machine learning literature, Such as: Correlation- based Feature Selection [6], Principal Component Analysis [6], Information Gain attribute evaluation [6], Gain Ratio attribute evaluation [6], Chi-Square Feature Evaluation [6] and Support Vector Machine feature elimination [5]. Some of these methods does not perform feature selection but only feature ranking, they are usually combined with another method when one needs to find out the appropriate number of features. Forward selection, backward elimination, bi-directional search, best-first search [12], genetic search [4], and other methods are often used on this task. The most often used criteria for feature selection is information theoretic based such as the Shannon entropy measure I for a dataset. The main drawback of the entropy measure is its sensitivity to the number of attribute values [11]. Therefore C4.5 uses gain ratio. However, this measure suffers the drawback that it may choose attributes with very low information content [9]. A comprehensive discussion on Bayes theorem for feature selection is available in [1].

3 Problem Statement

Feature selection is the process of removing features from the dataset that are irrelevant with respect to the task that is to be performed. Feature selection can be extremely useful in reducing the dimensionality of the data to be processed by the classifier, reducing execution time and improving predictive accuracy. This paper presents a novel algorithm for feature selection. Bayes theorem is used to discover dependency information among features. The basic idea of the algorithm is to test the

dependency of all pairs of attributes in deciding the value of the class attribute. The dependency of two attributes is measured by the conditional probabilities of the class attribute given the values of the attributes, which can be computed by Bayes theorem. In addition to that, feature selection has been improvised by Information gain which selects features based on their importance. The objective is to improve the classification accuracy such that prediction of the class variable is improved over that of the original data with initial attribute set and also reduces the computational time.

4 Proposed Method

We are concerned with the problem of feature selection. The main idea provided is to find out the dependent features and remove the redundant ones among them. The technology to obtain the dependency needed is based on Bayes theorem. The purpose of the proposed method is to reduce the computational complexity and increase the classification accuracy of the selected feature subsets. The dependence between two attributes is determined based on the probabilities of their joint values that contribute to positive and negative classification decisions. If there is an opposing set of attribute values that do not lead to opposing classification decisions (zero probability), then the two attributes are considered independent, otherwise dependent. One of them can be removed using Information gain measure and thus the number of attributes is reduced. A new feature selection method using Bayes theorem and Information gain is implemented and evaluated through extensive experiments, comparing with traditional feature selection algorithms over fifteen datasets from UCI machine learning repository databases[2]. The main steps of the proposed algorithm are given below.

1) Let $A = \{a_1, a_2, a_3, \dots, a_n\}$ be the initial set of attributes and $a_1 = \{a_{11}, a_{12}, \dots, a_{1n}\} \dots a_n = \{a_{n1}, a_{n2}, \dots, a_{nn}\}$; Group attributes in set A into an attribute set of pairs.

2) For each attribute pair (X_i, X_j)

Calculate $P[C_k / (X_i, X_j)]$ and then $P[C / (X_i, X_j)]$

using Bayes theorem

$$P[C_k / (X_i, X_j)] = P[(X_i, X_j) / C_k] * P(C_k)$$

$$P[C / (X_i, X_j)] = P[C_1 / (X_i, X_j)] + P[C_2 / (X_i, X_j)] + \dots + P[C_n / (X_i, X_j)]$$

If $P[C / (X_i, X_j)] > Threshold$ then X_i and X_j are dependent. Find the most relevant attribute using Information gain and keep it in the attribute list and remove the other one.

3) For each attribute in the list

Find gain value

If it exceeds the threshold retain it.

Else remove it.

4) End.

The proposed algorithm is enumerated as follows. Consider a weather dataset with the following features: Outlook, Temperature, Humidity and Windy is shown in Table 1. The class attribute play says whether a person can play or not using values of the predictive attributes.

Let $A = \{\text{Outlook, Temperature, Humidity, Windy}\}$,

Outlook= {Sunny, Over, Rainy},

Temperature= {Hot, Mild, Cool},

Humidity= {High, Normal}

Windy= {True, False}

And

Class attribute Play= {Yes, No}.

Attribute pairs are formed such as

{
 (Outlook, Temperature),
 (Outlook, Humidity),
 (Outlook, Windy),
 (Temperature, Humidity),
 (Temperature, Windy),
 (Humidity, Windy) }

For the attribute pair (Outlook, Temperature) the dependency is calculated as follows

$$P(\text{sunny, hot/no}) = 4/4 = 1.0000$$

$$P(\text{over, mild/yes}) = 1/6 = 0.1667$$

$$P(\text{over, cool/yes}) = 1/6 = 0.1667$$

$$P(\text{rainy, cool/yes}) = 4/6 = 0.6667$$

$$P(\text{yes}) = 6/10 = 0.6$$

$$P(\text{no}) = 4/10 = 0.4$$

$$P(X_i, X_j / \text{yes}) = 0.1667 * 0.1667 * 0.6667 = 0.0185$$

$$P(\text{yes} / X_i, X_j) = P(X_i, X_j / \text{yes}) * P(\text{yes}) \\ = 0.0185 * 0.6 = 0.0111$$

$$P(X_i, X_j / \text{no}) = 1.000$$

$$P(\text{no} / X_i, X_j) = P(X_i, X_j / \text{no}) * P(\text{no}) \\ = 1.000 * 0.4 = 0.4000$$

$$P(C / X_i, X_j) = P(\text{yes} / X_i, X_j) + P(\text{no} / X_i, X_j) \\ = 0.0111 + 0.4000 = 0.4111$$

For the attribute pair (Outlook, Humidity) the dependency is calculated as follows

$$\begin{aligned}
 P(\text{sunny, high/no}) &= 3/4 = 0.750 \\
 P(\text{sunny, normal/no}) &= 1/4 = 0.250 \\
 P(\text{over, high/yes}) &= 1/6 = 0.167 \\
 P(\text{over, normal/yes}) &= 1/6 = 0.167 \\
 P(\text{rainy, high/yes}) &= 1/6 = 0.167 \\
 P(\text{rainy, normal/yes}) &= 3/6 = 0.500 \\
 P(\text{yes}) &= 6/10 = 0.6 \\
 P(\text{no}) &= 4/10 = 0.4 \\
 P(X_i, X_j / \text{yes}) &= 0.167 * 0.167 * 0.167 * 0.500 = 0.0023 \\
 P(\text{yes} / X_i, X_j) &= P(X_i, X_j / \text{yes}) * P(\text{yes}) \\
 &= 0.0023 * 0.6 = 0.0013 \\
 P(X_i, X_j / \text{no}) &= 0.750 * 0.250 = 0.1875 \\
 P(\text{no} / X_i, X_j) &= P(X_i, X_j / \text{no}) * P(\text{no}) \\
 &= 0.1875 * 0.4 = 0.075 \\
 P(C / X_i, X_j) &= P(\text{yes}/X_i, X_j) + P(\text{no}/X_i, X_j) \\
 &= 0.0013 + 0.0750 = 0.0763
 \end{aligned}$$

Similarly,

$$\begin{aligned}
 &\text{For the pair (Outlook, Windy)} \\
 P(C / X_i, X_j) &= 0.0139 \\
 &\text{For the pair (Temperature, Humidity)} \\
 P(C / X_i, X_j) &= 0.0026 \\
 &\text{For the pair (Temperature, Windy)} \\
 P(C / X_i, X_j) &= 0.0034 \\
 &\text{For the pair (Humidity, Windy)} \\
 P(C / X_i, X_j) &= 0.0291
 \end{aligned}$$

From the above observation, we found that there exists a dependency between the attribute Outlook and Temperature. Any one attribute is removed based on the gain value.

5 Experimental Results and Discussion

The proposed feature selection using Bayes theorem and Information gain is applied to many datasets, and the performance evaluation is done using a software package called WEKA [10]. We presented the performance evaluation on fifteen dataset such as Contactlenses, Shuttlelanding, DNAPrometer, TicTocToe, Parity, Nursery, Adult, Chess, Monk, Weather, Splice, Spectheart, King-Rook vs. King-Pawn, Car-evaluation and Balloon. All these datasets are recommended by UCI repository databases [2]. A summary of dataset is presented in Table 2.

For each dataset, we run seven feature selection methods such as BT-IG, CFS subset evaluation, Chi-square attribute evaluation, Gain ratio, Information Gain, one attribute

evaluation and symmetrical uncert attribute evaluation respectively, and record the running time and the number of selected features for each algorithm. We then apply ID3, C4.5 and NB on the original dataset (See Table 3) as well as each newly obtained dataset containing only the selected features from each algorithm (See Table 4 and 8) and recorded the overall accuracy using 10 fold cross- validation (See Table 5-8).A new feature selection algorithm using Bayes theorem and Information gain is implemented and evaluated through extensive experiments comparing with related feature selection algorithms.Our findings can be summarized as follows:

- (i). We have implemented a new feature selector using Bayes theorem and Information gain and found that it performs better than the popular and computationally more expensive traditional algorithms by improving the performance of classifiers such as NB, C4.5 and ID3.
- (ii). Improved NB classifier performance from 83.93 to 86.00 %, where 83.93 indicates the performance using Information gain.
- (iii). Improved C4.5 classifier performance from 79.36 to 82.17 %, where 79.36 indicates the performance using Information gain.
- (iv). Improved ID3 classifier performance from 83.43 to 84.11, where 83.43 indicates the performance using Information gain.
- (v). Reduced the number of selected features when compared to other feature selection methods in the datasets such as Contact lenses, Nursery, Monk, Parity, Tic-Tac and Weather.
- (vi). Improved performance when compared to the performance of conventional algorithms, With C4.5 classifier in datasets such as Contact lenses, Shuttle, Tic-Tac, Nursery, Monk and Car. With ID3 classifier in the dataset such as Contact lenses, Tic- Tac, Nursery, Monk, Weather and Car. With NB classifier in the dataset such as Contact lenses,Shuttle,Tic-Tac,Nursery,Monk,Weather and Car.

Fig 1. shows the performance results of conventional and proposed algorithm. From Fig 1, it is found that, when selected attributes by the proposed method are used for the classification, the classification accuracy increases significantly.

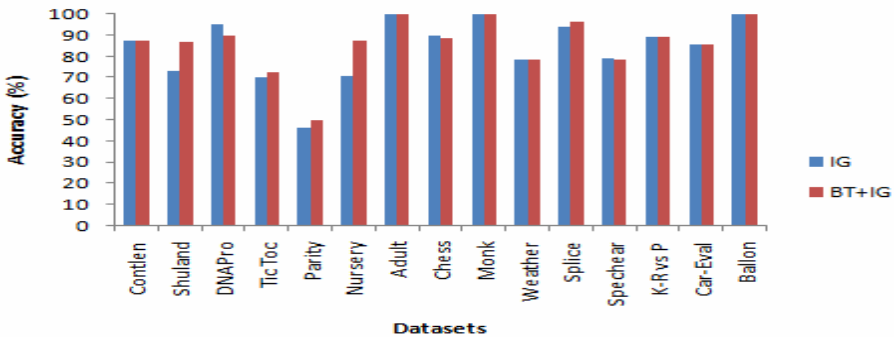


Fig. 1. Accuracy of NB classifier on selected features by IG and BT+IG

Table 1. A Sample Dataset used in the experiment

OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Over	Mild	High	False	Yes
Rainy	Cool	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	Yes
Over	Cool	Normal	True	Yes
Sunny	Hot	High	False	No
Sunny	Hot	Normal	False	No
Rainy	Cool	Normal	False	Yes

Table 2. Details description of dataset used in the experiment

Datasets	Instances	Attributes
Contactlense	24	5
Shuttlelanding	15	7
DNAprometer	106	58
TicTocToe	958	10
Parity	100	11
Nursery	12960	9
Adult	20	5
Chess	2128	37
Monk	124	6
Weather	14	5
Splice	3190	61
Spectheart	267	23
King-Rook vs King-Pawn	3196	37
Car-evaluation	1728	7
Balloon	20	5

Table 3. Accuracy of Classifiers on full feature set

Datasets	NB	ID3	J48
Contactlense	70.83	70.83	83.33
Shuttlelanding	80.00	60.00	53.33
DNAPrometer	90.57	76.42	81.13
TicTocToe	69.62	85.07	83.40
Parity	40.00	45.00	44.00
Nursery	90.32	98.18	97.05
Adult	100.00	100.00	100.00
Chess	89.94	99.20	98.91
Monk	99.36	89.53	94.36
Weather	57.14	85.71	50.00
Splice	95.36	89.53	94.36
Speckheart	79.03	70.04	80.90
King Rook vs King Pawn	87.89	99.68	99.40
Car-evaluation	85.53	89.35	92.36
Balloon	100.00	100.00	100.00
Mean	82.37	83.90	73.87

Table 4. Number of selected features by each feature selection algorithm

Datasets	CFS subset eval	Chisquare attr eval	Gain ratio	Information gain	One attr eval	Symmetrical uncert attr eval
Contactlense	1	2	2	2	3	1
Shuttlelanding	2	6	3	5	4	4
DNAPrometer	6	6	6	6	5	5
TicTocToe	5	1	1	1	1	1
Parity	3	6	6	6	6	6
Nursery	1	1	1	1	5	5
Adult	2	2	2	2	2	2
Chess	6	7	5	7	5	9
Monk	2	2	2	2	2	2
Weather	2	2	2	2	3	2
Splice	22	7	8	8	9	2
Speckheart	12	8	10	9	16	9
King-Rook vs King-Pawn	7	11	15	11	19	10
Car-evaluation	1	6	6	6	6	6
Balloon	2	2	2	2	2	2

Table 5. Accuracy of NB on selected features for each feature selection algorithm

Datasets	CFS subset eval	Chisquare attr eval	Gain ratio	Information gain	One attr eval	Symmetrical uncert attr eval
Contactlense	70.83	87.50	87.50	87.50	54.17	70.83
Shuttlelanding	80.00	73.33	80.00	73.33	73.33	73.33
DNAprometer	95.28	95.28	95.28	95.28	95.28	95.34
TicTocToe	72.44	69.94	69.94	69.94	69.94	69.94
Parity	50.00	46.00	46.00	46.00	47.00	46.00
Nursery	70.97	70.97	70.97	70.97	88.84	70.97
Adult	100.00	100.00	100.00	100.00	100.00	100.00
Chess	94.45	89.61	92.34	89.61	86.33	90.23
Monk	100.00	100.00	100.00	100.00	100.00	100.00
Weather	78.57	78.57	78.57	78.57	71.43	78.57
Splice	96.14	93.89	94.17	94.17	94.29	94.17
Spectheart	82.02	76.78	80.15	79.03	79.03	79.03
King-Rook vs King-Pawn	91.99	88.17	89.86	89.11	88.11	88.67
Car-evaluation	70.02	85.53	85.53	85.53	85.53	85.53
Balloon	100.00	100.00	100.00	100.00	100.00	100.00
Mean	83.51	83.70	84.68	83.93	82.21	82.84

Table 6. Accuracy of ID3 on selected features for each feature selection algorithms

Datasets	CFS subset eval	Chisquare attr eval	Gain ratio	Information gain	One attr eval	Symmetrical uncert attr eval
Contactlense	70.83	87.50	87.50	87.50	50.00	70.83
Shuttlelanding	46.67	60.00	66.67	66.67	66.67	66.67
DNAprometer	84.91	84.91	84.91	84.91	84.91	84.91
TicTocToe	82.78	69.94	69.93	69.94	69.94	69.94
Parity	53.00	53.00	53.0000	53.00	48.00	53.00
Nursery	70.97	70.97	70.9722	70.97	91.70	70.97
Adult	100.00	100.00	100.0000	100.00	100.00	100.00
Chess	94.36	94.36	92.3402	94.36	90.60	94.36
Monk	95.97	95.97	95.9677	95.97	95.97	95.97
Weather	78.57	78.57	78.5714	78.57	57.14	78.57
Splice	90.66	90.60	90.3135	90.31	88.87	90.31
Spectheart	81.65	79.40	75.6554	75.66	79.03	75.66
King-Rook vs King-Pawn	94.24	96.09	94.7434	94.34	96.18	94.24
Car-evaluation	70.02	89.35	89.3519	89.35	89.35	89.35
Balloon	100.00	100.00	100.0000	100.00	100.00	100.00
Mean	80.97	83.37	83.32	83.43	80.55	82.31

Table 7. Accuracy of C4.5 on selected features for each feature selection algorithms

Datasets	CFS subset eval	Chisquare attr eval	Gain ratio	Information gain	One attr eval	Symmetrical uncert attr eval
Contactlense	70.83	87.50	87.50	87.50	58.33	70.83
Shuttlelanding	53.33	53.33	60.00	53.33	60.00	53.33
DNAPrometer	83.02	83.02	83.02	83.02	83.02	83.96
TicTocToe	79.44	69.94	69.94	69.94	69.94	69.94
Parity	44.00	40.00	40.00	40.00	50.00	40.00
Nursery	70.97	70.97	70.97	70.97	90.74	70.97
Adult	100.00	100.00	100.00	100.00	100.00	100.00
Chess	94.31	94.36	92.34	94.36	90.60	94.31
Monk	91.94	91.94	91.94	91.94	91.94	91.94
Weather	42.86	42.86	42.86	42.86	50.00	42.86
Splice	94.48	93.54	94.01	94.01	93.98	94.01
Spectheart	81.65	79.40	75.66	75.66	79.03	75.66
King-Rook vs King-Pawn	94.06	96.50	94.71	94.49	96.81	94.06
Car-evaluation	70.02	92.36	92.36	92.36	92.36	92.36
Balloon	100.00	100.00	100.00	100.00	100.00	100.00
Mean	78.06	79.71	79.68	79.36	80.44	78.28

Table 8. Number of selected features and accuracy of NB, ID3 and C4.5 on selected features for proposed method

Datasets	Total features	Selected features	NB	ID3	C4.5
Contactlense	5	2	87.50	87.50	87.50
Shuttlelanding	7	3	86.67	46.67	53.33
DNAPrometer	58	26	89.62	77.36	81.13
TicTocToe	10	5	72.44	82.78	79.44
Parity	11	4	50.00	50.00	44.00
Nursery	9	4	87.59	90.14	90.14
Adult	5	2	100.00	100.00	100.00
Chess	37	13	88.58	97.32	96.62
Monk	6	2	100.00	95.97	91.94
Weather	5	2	78.57	78.57	42.86
Splice	61	19	96.11	91.41	94.48
Spectheart	23	18	78.65	73.03	81.27
King-Rook vs King-Pawn	37	14	89.05	96.70	96.71
Car-evaluation	7	5	85.36	94.21	93.23
Balloon	5	2	100.00	100.00	100.00
Mean	19	8	86.00	84.11	82.17

6 Conclusion

Selecting the right set of features for classification is one of the most important problems in designing a good classifier. Decision Tree induction algorithms such as C4.5 have incorporated in their learning phase an automatic feature selection strategy while some other statistical classification algorithm require the feature subset to be selected in a pre-processing phase. It is well known that correlated and irrelevant features may degrade the performance of the classification algorithms. In our study, we evaluated the influence of feature pre-selection on the predictive accuracy of DT's and NB classifiers using the real world dataset. We observed that the accuracy of the C4.5 and NB classifiers could be improved with an appropriate feature pre-selection phase for the learning algorithm. Beyond that, the number of features used for classification can also be reduced since feature selection is a time consuming process.

References

- [1] Balamurugan, S.A., Rajaram, R.: Effective and Efficient Feature Selection for Large Scale Data using Bayes Theorem. *Journal of Automation and Computing* 6(1), 62–71 (2009)
- [2] Blake, C.L., Merz, C.J.: *UCI Repository of Machine Learning Databases* (2008), <http://www.ics.uci.edu/~mllearn/mlrepository.html>
- [3] Cover, T.M.: On the possible ordering on the measurement selection problem. *IEEE Transactions on SMC* 7(9), 657–661 (1977)
- [4] Goldberg, D.E.: *Genetic algorithms in search, optimization and machine learning*. Addison-Wesley, Reading (1989)
- [5] Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. *Machine Learning* 46, 389–422 (2002)
- [6] Hall, M.A., Smith, L.A.: Practical feature subset selection for machine learning. In: *Proceedings of the 21st Australian Computer Science Conference*, pp. 181–191 (1998)
- [7] Han, J., Kamber, M.: *Data mining Concepts and Techniques*. Morgan Kaufmann, San Francisco (2006)
- [8] Kohavi, R., John, G.H.: The Wrapper approach. In: Lui, H., Matoda, H. (eds.) *Feature Extraction Construction and Selection*, pp. 30–47. Kluwer Academic Publishers, Dordrecht (1998)
- [9] Lopez de Mantaras, R.: A Distance- based attribute selection measure for decision tree induction. *Machine Learning* 6, 81–92 (1991)
- [10] WEKA, Open Source Collection of Machine Learning Algorithms
- [11] White, A.P., Lui, W.Z.: Bias in the information- based measure in decision tree induction. *Machine Learning* 15, 321–329 (1994)
- [12] Witten, H., Frank, E.: *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd edn. Morgan Kaufmann, San Francisco (2005)

Data Mining Technique for Knowledge Discovery from Engineering Materials Data Sets

Doreswamy¹, K.S. Hemanth², Channabasayya M. Vastrad³, and S. Nagaraju⁴

^{1,2} Department of Computer Science
Mangalore University, Mangalagangothri-574 199, Karnataka, India
doreswamyh@yahoo.com, reachhemanthmca@gmail.com

³ Department Computer Science & Engineering
PDIT, Hospet, Karnataka, India
chennu.vastrad@gmail.com

⁴ Department Computer Science & Engineering
Bahubali College of Engineering, Shravanabelagola,
Hassan-573135, Karnataka, India
nagaraju.sms@gmail.com

Abstract. The goal of this paper is to discuss how data mining technique can be applied in materials informatics to extract knowledge from materials data. Studying material data sets from a data mining perspective can be beneficial for manufacturing and other industrial engineering applications. This work employs an effective materials classification system on design requirements. Experiments were conducted on material datasets that consist of all class of materials. The algorithm of the Naive Bayesian classifier is implemented successfully enabling it to solve classification problems and the outcomes can be very useful for design engineers to speed up decision making process in manufacturing and other industrial engineering applications. The comparison of performance with various domains of material classes confirms the advantages of successive learning and suggests its application to other learning domains.

Keywords: Knowledge Discovery, Materials informatics, Naive Bayesian Classifier.

1 Introduction

The rapid developments in materials science and information technologies have influenced the large volume of massive data sets and materials informatics respectively. Materials informatics a field of study that applies the principles of informatics to materials science and engineering to better understand the use, selection, development, and discovery of materials.

As a lot of traditional analytic techniques employed for materials structural-properties analysis and not effective any longer under these situations, researchers in the manufacturing industries and other industrial engineering applications areas are being faced more new research issues in systematic analysis of materials data sets. Therefore, materials informatics has been emerging in material science and technology as a new

research areas[5],[10],[11], and has already changed the experimental methods and way of thinking in materials research, and will lead even more challenges in interdisciplinary research.

Data Mining is an interdisciplinary field merging ideas from statistics, machine learning, information science, visualization and other disciplines[7]. It is a very useful approach to integrate information and theory for knowledge discovery from any informatics such as Bioinformatics, Chemoinformatics, Nano informatics, Materials informatics and so on. The impact of Data Mining and knowledge discovery has been evidenced by many successful research experimental results[19],[20],[21],[22]. Therefore, Data mining can be used to extract non-trivial, hidden, potential useful and ultimately understandable knowledge from massive materials databases[29],[30].

Data Mining has two primary Models: Descriptive Data Mining Model and predictive Data Mining Model. Descriptive mining models describe or summarize the general characteristics or behaviour of the data in the materials database. Predictive models perform inference on the current data in order to make the prediction. Both of them are fundamentals to understand materials behaviors. In general, in materials informatics, Data mining can be used in the following task[13]:

- (i). **Association analysis:** Association analysis is good at discovering patterns, and can be used to develop heuristic rules for materials behaviour based on large data sets[26],[27],[28].
- (ii). **Classifier/Predict modelling:** Some machine learning algorithms can be used for materials class prediction and materials classification models such as support vector regression (SVR) and neural network (NN), can be used to build up the Predict models[31]. These models can be used to predict crystal structure or composite materials properties from fused materials data[14].
- (iii). **Cluster analysis:** As an exploratory data analysis tool, it can sort different materials or properties into groups in such a way that the degree of association between two objects is maximal if they belong to the same group and minimal otherwise. And, cluster analysis can be integrated with high-throughput experimentation for rapidly screening combinatorial data[20].
- (iv). **Outlier Analysis:** In properties analysis or combinatorial experiments, outlier analysis is used to identify anomalies, especially to assess the uncertainty and accuracy of results, and distinguish between true discoveries and false-positive results.
- (v). **Material visualization:** Reconstruction of material structure information based on materials data would help researchers to analyze the relationships between material structure and material properties[16].

The rest of the paper is organized as follows: scope of knowledge discovery on materials informatics is discussed in section 2. The section 3 describes naive Bayesian classifier algorithm and performance measures. The experimental results are presented in the section 4. The conclusions and future scopes are given in the section 5.

2 Scope of Data Mining in Materials Informatics

Data Mining is becoming an increasingly valuable tool in the broad area of materials development and design[4],[9], and there are good reasons why this area is particularly

rich for materials informatics[14],[15],[16],[17]. There is a massive range of possible new materials, and it is often complex to physically model the relationships between constituents, processing and final properties. Therefore, materials are primarily still developed by quantitative and trial-and-error procedures, where researchers are guided by experience and heuristic rules for materials classification, selection and property predictions. These rules are applied to somewhat limited materials data sets of constituents and processing conditions, but then try as many combinations as possible to find materials with desired properties. This is essentially human Data Mining, where one's brain, rather than the computer, is being used to find correlations, make predictions, and design optimal strategies. Transferring Data Mining tasks from human to computer offers the potential to enhance accuracy, handle more data, and to allow wider dissemination of accrued knowledge[5].

Materials informatics[18],[19] has been a subject of materials science, since the international conference of "Materials Informatics-Effective Data Management for New Materials Discovery" was held in Boston in 1999. Wei[24] described that materials informatics is a new subject that leverages information technology and computer network technology to represent, parse, store, manage and analyze the material data, in order to realize the sharing and knowledge mining of materials data for uncovering the essence of materials, and accelerate the new material discovery and design. The research areas of materials informatics are mainly focused on following tasks[25]:

- i. **Data standards:** There are thousands of materials databases in different formats[32], and they are difficult to communicate with each other. To standard these databases and to integrate materials data into a single or coherent database, data pre-processing is the first important task of materials informatics[23] to enable knowledge discovery.
- ii. **Organization and management of material data:** In order to meet materials researchers' different needs, satisfy the need of research and production, to construct the materials data into a whole and single coherent database, efficient Materials Database Management Systems(MDBMS)is very necessary[25].
- iii. **Data mining on materials data:** There is an enormous range of possible new materials, and it is often difficult to physically model the relationships between constituents, and processing, and final properties. Data mining has the abilities to search, classify, select and analyze material data and to find potential, previously unknown patterns rules. It involves selecting, exploring and modelling large amounts of data to uncover previously unknown patterns from large materials databases[7]. Data mining involves some high-effective computational algorithms[18],[19], such as neural networks, genetic algorithm, etc.

3 Data Mining Technique

Classification and prediction is one of the core tasks of Data Mining. A classification technique is a systematic approach to building classification models from input data set. Several classification models are reported in literature such include Decision Tree Classifier, Rule-Based Classifier, Neural Network Classifier, naive Bayesian Classifier, Neuro-fuzzy classifier, Support Vector Machines and etc. Each technique employs a

learning algorithm to identify a model that best fits the relationships between the attribute set and class label of the input data. The model generated by a learning algorithm should both fit the input data well and correctly predict the class labels of data set that has never seen before. Therefore, the key objective of the learning algorithm is to build models with good generalization capacity.

3.1 Naive Bayesian Classifier

Naive Bayesian classifier is a statistical classifier that can predict class membership probabilities such as the probability that a given tuple belongs to a particular class. It is fast and incremental that can deal with discrete and continuous attributes and has excellent performance in real-life problems. The naive Bayesian classifier, or simple Bayesian classifier generally used for classification or prediction task. As it is simple, robust and generality, this procedure has been deployed for various applications such as Materials damage detection[1],[2], Agricultural land soils classification[6], Network intrusion detection[8], Machine learning applications[19]. Therefore, the application of this method is extended to classification of engineering materials data sets[4],[6],[9] and to reduce the computational cost of classification of materials. The working procedure of the naive Bayesian classifier is shown in the followings section.

3.2 Algorithm of Naive Bayesian Classifier

1. Let D be a training set of tuples and their associated class labels. Each tuple is represented by an n -dimensional attribute vector, $X = (x_1, x_2, \dots, x_n)$, depicting n measurements made on the tuple from n attributes, respectively, A_1, A_2, \dots, A_n .
2. Suppose that there are m classes, C_1, C_2, \dots, C_m . Given a tuple, X , the classifier will predict that X belongs to the class having the highest posterior probability, conditioned on X . That is, the naive Bayesian classifier predicts that tuple X belongs to the class C_i , if and only if

$$P(C_i/X) > P(C_j/X) \text{ for all } 1 \leq j \leq m; j \neq i. \quad (1)$$

3. Thus it maximizes $P(C_i/X)$. The class C_i for which $P(C_i/X)$ is maximized is called the maximum posteriori hypothesis. By Bayes' theorem.

$$P(C_i/X) = \frac{P(X/C_i)P(C_i)}{P(X)} \quad (2)$$

As $P(X)$ is constant for all classes, only $P(X/C_i)P(C_i)$ need be maximized. If the class prior probabilities are not known, then it is commonly assumed that the classes are equally likely, that is, $P(C_i) = P(C_2) = P(C_3) = \dots = P(C_m)$, and it would therefore maxi-

mize $P(X / C_i)$. Otherwise, it maximizes $P(X / C_i)P(C_i)$. Note that the class prior probabilities may be estimated by $P(C_i) = |C_i, D| / |D|$, where $|C_i, D|$ is the number of training tuples of class C_i in D .

- Given data sets with many attributes, it would be extremely computationally expensive to compute $P(X / C_i)$. In order to reduce computation in evaluating $P(X / C_i)$, the naive assumption of class conditional independence is made. This presumes that the values of the attributes are conditionally independent of one another, given the class label of the tuple (i.e., that there are no dependence relationships among the attributes). Thus,

$$\begin{aligned}
 P(X / C) &= \prod_{k=1}^n P(X_k / C_i) \\
 &= P(X_1 / C_i) \times P(X_2 / C_i) \times P(X_2 / C_i) \times \dots \times P(X_n / C_i) \quad (3)
 \end{aligned}$$

The probabilities $P(X_1 / C_i), P(X_2 / C_i), P(X_3 / C_i) \dots P(X_n / C_i)$ can easily be estimated from the training tuples. Recall that here x_k , refers to the value of attribute A_k , for tuple X . For each attribute, the attribute value may be either categorical or continuous-valued. For instance, to compute $P(X / C_i)$, it is considered the following:

If A_k is categorical, then $P(X_k / C_i)$ is the number of tuples of class C_i in D having the value for A_k , divided by $|C_i, D|$, the number of tuples of class C_i in D .

- In order to predict the class label of X , $P(X / C_i)P(C_i)$ is evaluated for each class C_i . The classifier predicts that the class label of tuple X is the class C_i if and only if

$$P(X / C_i)P(C_i) > P(X / C_j) P(C_j) \text{ for all } 1 \leq j \leq m; j \neq i \quad (4)$$

In other words, the predicted class label is the class C_i for which $P(X / C_i)P(C_i)$ is the maximum.

3.3 Performance Measures

The classifier in this research is evaluated on engineering materials data set using the standard metrics of accuracy, precision, and recall. These were calculated using the predictive classification table, known as Confusion Matrix[16].

Table 1. Confusion Matrix

		PREDICTED	
		IRRELEVANT	RELEVANT
ACTUAL	IRRELEVANT	TN	FP
	RELEVANT	FN	TP

Where:

TN (True Negative) : Number of correct predictions that an instance is irrelevant

FP (False Positive) : Number of incorrect predictions that an instance is relevant

FN (False Negative) : Number of incorrect predictions that an instance is irrelevant

TP (True Positive) : Number of correct predictions that an instance is relevant

Accuracy – The proportion of the total number of predictions that were correct:

$$\text{Accuracy (\%)} = (TN + TP) / (TN + FN + FP + TP) \quad (5)$$

Precision – The proportion of the predicted relevant materials data sets that were correct:

$$\text{Precision (\%)} = TP / (FP + TP) \quad (6)$$

Recall – The proportion of the relevant materials data sets that were correctly identified

$$\text{Recall (\%)} = TP / (FN + TP) \quad (7)$$

The classification performance of the naive Bayesian classifier is analyzed with standard metrics in the experimental results.

4 Experimental Results and Discussion

In this experiment, materials database is organized by sampling material data sets from peer-reviewed research papers published[23] and from poplar materials website <http://www.matweb.com>. The tuples of data table consists of both numerical and categorical attribute values. The tuples consisting only categorical attributes and their values are considered for finding probable class of materials. Atypical set of training sample data sets is shown in the following table 2.

A prototype software module realizing Naive Bayesian classifier is designed and developed using .NET technology as it is efficient for handling data objects. This

Table 2. Partial List Of Training Samples And Their Attribute Values

CR	CH	CE	SM	CAST	EXTRN	MANFT	CS	MACHN	FS	WA	Class Label
Excellent	Poor	NIL	Good	Fair	Good	Excellent	Poor	Good	Poor	Poor	P
Good	Poor	NIL	Good	Fair	Good	Excellent	Poor	Good	Poor	Poor	P
Good	Poor	NIL	Good	Fair	Good	Excellent	Poor	Good	Poor	Poor	P
Good	Poor	NIL	Good	Fair	Good	Excellent	Poor	Good	Poor	Poor	P
Very Good	Poor	NIL	Good	Fair	Good	Excellent	Poor	Good	Poor	Poor	P
Excellent	Poor	Good	Poor	Poor	Poor	Good	Excellent	Poor	Good	Poor	C
Excellent	Poor	Good	Poor	Poor	Poor	Good	Excellent	Poor	Good	Poor	C
Good	Poor	Good	Poor	Poor	Poor	Good	Excellent	Poor	Good	Poor	C
Good	Fair	Good	Poor	Poor	Poor	Good	Excellent	Poor	Good	Poor	C
Good	Fair	Good	Poor	Poor	Poor	Good	Excellent	Poor	Good	Poor	C
Poor	Very Good	Excellent	Excellent	Excellent	Excellent	Fair	Good	Good	Good	Poor	M
Poor	Good	Excellent	Excellent	Excellent	Excellent	Fair	Good	Good	Good	Poor	M
Good	Good	Excellent	Excellent	Excellent	Excellent	Fair	Good	Good	Good	Poor	M
Fair	Good	Excellent	Excellent	Excellent	Excellent	Fair	Good	Good	Good	Poor	M
Poor	Good	Excellent	Excellent	Excellent	Poor	Fair	Good	Good	Excellent	Fair	M
Good	Fair	Good	Poor	Poor	Fair	Good	Excellent	Poor	Good	Fair	C
Good	Fair	Good	Poor	Poor	Fair	Good	Excellent	Poor	Good	Fair	C
Poor	Very Good	Good	Excellent	Excellent	Very Good	Fair	Good	Good	Good	Good	M
Poor	Good	Good	Excellent	Excellent	Good	Fair	Good	Good	Poor	Good	M
Good	Poor	Good	Good	Fair	Good	Excellent	Poor	Good	Poor	Fair	P

CR: Chemical Resistance, CH: Conductivity-Heat, CE: Conductivity-Electricity SM: Sheet Metal, CAST: Casting, EXTRN: Extrusion, MANFT: Manufacturing, CS: Creep Strength, MACHN: Mach inability, FS: Fatigue Strength, WA: Water Absorptions

software module accepts design requirements from the user's Graphical User Interface(GUI) and predicts probable class to which design requirements belong. The design requirements associated geometrical features of the material are determined by design engineers or through CAD systems. The GUI of the implemented software module is shown in the following figure 1.

The Naive Bayesian classifier is trained on engineering materials data set consisting of 1630 data sets and these data sets consist of 15 discrete and categorical attributes. A sample data set is randomly selected from the testing data set and input to the classifier, then classifier predicts the probable knowledge, class of the input data set. Misclassification occurs when an input data sample's categorical attribute values neither associated to any of the class of materials. The classification performance of naive Bayesian classifier is analyzed with the standard measures, which are used for measuring the other classifiers, shown in the figures 2 and 3. From the figure 2, the False Negative(FN) and False Positive(FP) measures in all the class of materials are very less that indicate the false data sets correctly classified by the classifier.

The knowledge extracted by the classifier includes the general features of the tested data sets, total number of data sets undergone for testing, number of data sets classified to each class, number of true data sets positively classified, number of true data sets negatively classified, number of false data sets positively classified and number of false data sets negatively classified, are visualized in the figure 2.

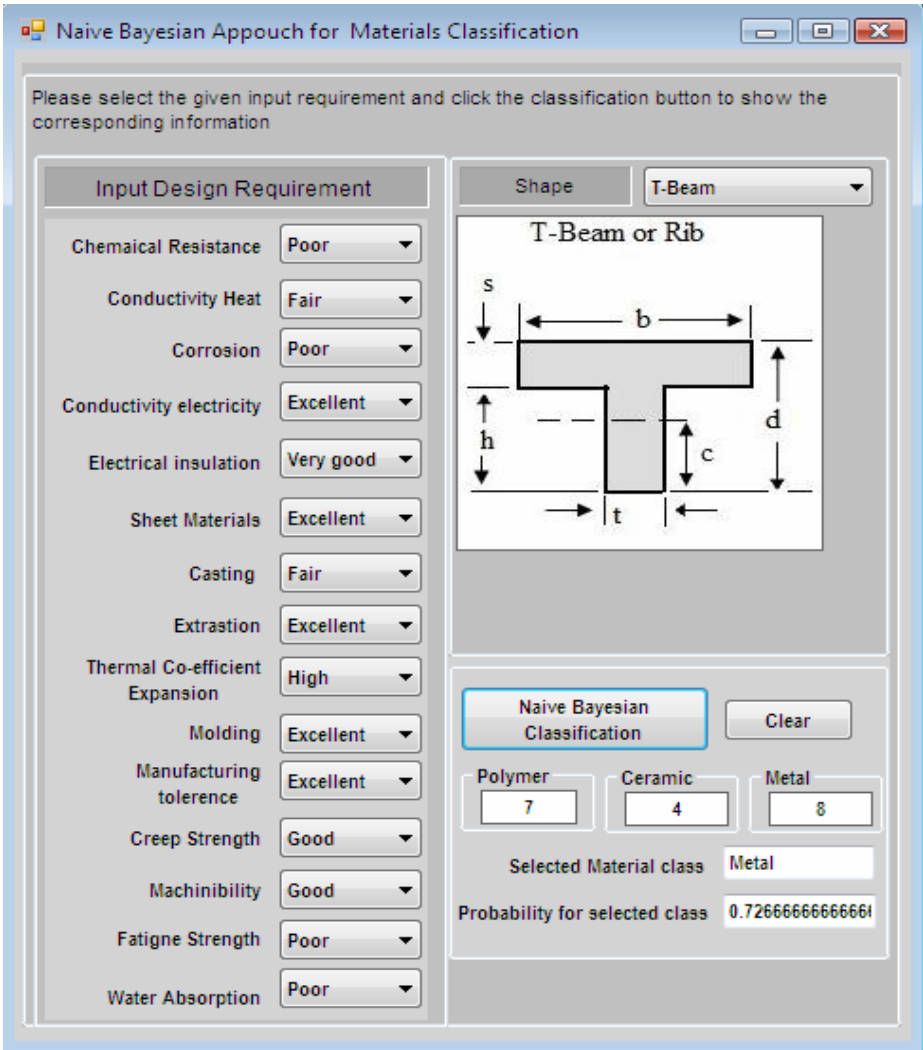


Fig. 1. A prototype software module for probable material's class prediction on input design requirements

The aggregated features computed with obtained general features include the performance measures such as Accuracy(ACC), Precision(PREC) and Recall (REC) and these are depicted in figure 3. ACC, PREC and REC are computed on 1630 data sets. Out these, ACC is 80.91% , PREC is 75.07% and REC is 94.37 % for 550 polymer materials. **ACC is 94.22%**, PREC is 93.98% and REC is 95.78% for 467 Ceramic materials, and **ACC is 95.43%**, PREC is 97.56% and REC is 95.69% for 617 metals materials.

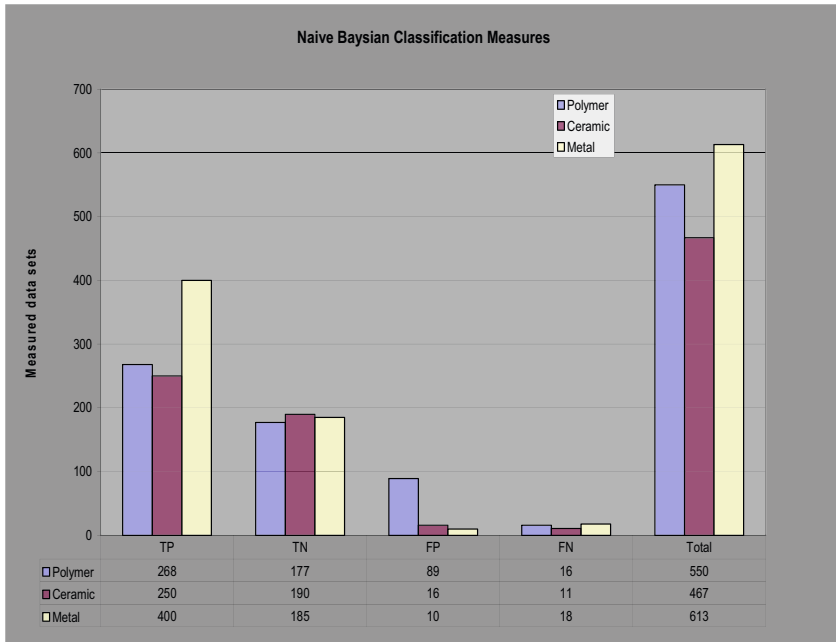


Fig. 2. Classification measures of TP,TN,FP,FN and total number of materials in each class

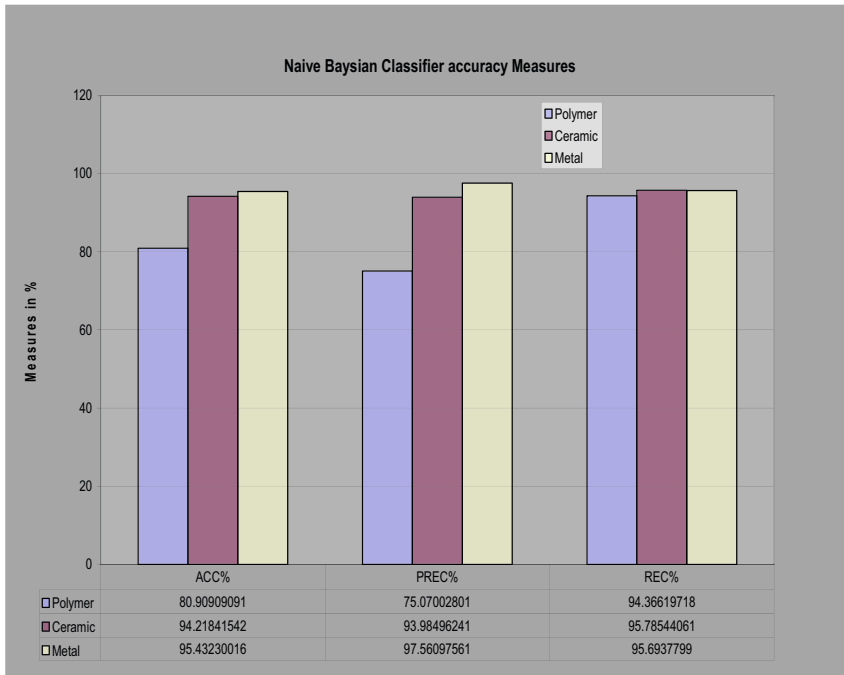


Fig. 3. Classification performance measures ACC, PREC, and REC

5 Conclusions and Future Scope

In this paper, Data Mining technique is applied in materials informatics to extract knowledge from materials data. The performance of the naive Bayesian classifier is analyzed on materials data sets. Studying material data sets from a data mining perspective can be beneficial for manufacturing and other industrial engineering applications. The algorithm of the Naive Bayesian classifier is applied successively enabling it to solve classification problems and the outcomes can be very useful for the manufacturing and other industrial engineering applications. The comparison of performance in various domains of material classes confirms the advantages of successive learning and suggests its application to other learning algorithms.

Further, an application of this algorithm can be extended to the classification of engineering materials data sets consisting of both numerical and categorical attribute values. Performance comparison of this algorithm with other classification algorithms on materials informatics data sets is the future scope of this research.

Acknowledgement. The authors wish to acknowledge the financial support from the University Grant Commission(UGC), INDIA for the Major Research Project “**Scientific Knowledge Discovery Systems (SKDS) For Advanced Engineering Materials Design Applications**” vide reference F.No. 34-99\2008 (SR), 30th December 2008, and also gratefully acknowledge the unanimous reviewers for their kind suggestions and comments for improving this paper.

References

- [1] Addin, O., Sapuan, S.M., Mahdi, E., Othman, M.: A Naive-Bayes classifier for damage detection in engineering, materials. *Materials and Design*, 2379–2386 (2007)
- [2] Addina, A.O., Sapuanb, S.M., Othmanc, M.: A Naïve-Bayes Classifier And F-Folds Feature Extraction Method For Materials Damage Detection. *International Journal of Mechanical and Materials Engineering (IJMME)* 2(1), 55–62 (2007)
- [3] Bhargavia, P., Jyothi, S.: Applying Naive Bayes Data Mining Technique for Classification of Agricultural Land Soils. *International Journal of Computer Science and Network Security* 9(8), 117–122 (2009)
- [4] Doreswamy, Sharma, S.C.: An Expert Decision Support System for Engineering Materials Selections And Their Performance Classifications on Design Parameters. *International Journal of Computing and Applications (ICJA)* 1(1), 17–34 (2006)
- [5] Doreswamy: A survey for Data Mining framework for polymer Matrix Composite Engineering materials Design Applications. *International Journal of Computational Intelligence Systems (IJCIS)* 1(4), 312–328 (2008)
- [6] Doreswamy: Engineering Materials Classification Model- A Neural Network Application. *IJDCDIS A Supplement, Advances in Neural Networks* 14(S1), 591–595 (2007)
- [7] Han, J., Kamber, M.: *Data Mining Concepts and Techniques*. Morgan Kaufmann Publisher, San Francisco (2009)
- [8] Khor, K.-C., Ting, C.-Y., Amnuaisuk, S.-P.: From Feature Selection to Building of Bayesian Classifiers: A Network Intrusion Detection Perspective. *American Journal of Applied Sciences* 6(11), 1949–1960 (2009)
- [9] Langseth, H., Nielsen, T.: Classification using Hierarchical Naïve Bayes models. *Machine Learning* 63(2), 135–159 (2006)

- [10] Chikyow, T.: Trends In Materials Informatics In Research On Inorganic Materials. *Quarterly Review* 20, 59–71 (2006)
- [11] Qunyi, W., Xiaodong, P., Xiangguo, L., Weidong, X.: Materials informatics and study on its further development. *Chinese Science Bulletin* 51(4), 498–504 (2006)
- [12] Callister, W.D.: *Materials Science and Engineering*. Wiley India Pvt. (2007)
- [13] Suh, C., Rajan, K.: Data mining and informatics for crystal chemistry: establishing measurement techniques for mapping structure-property relationships. *Materials Science And Technology* 25, 466–471 (2009)
- [14] Scott, D. J., Coveney, P.V., Kilner, J. A., Rossiny, J., Alford, N.: Prediction of the functional properties of ceramic materials from composition using artificial neural networks. *Journal Of The European Ceramic Society* 27, 4425–4435 (2007)
- [15] Ferris, F., Peurrung, L.M., Marder, J.: Materials informatics: Fast track to new materials. *Advanced Materials & Processes* 165, 50–51 (2007)
- [16] Yu., G., Chen., J., Zhu, L.: Data mining techniques for materials informatics: Datasets Preparing and Applications. In: *Proc. 2009 Second International Symposium on Knowledge Acquisition and Modeling*, pp. 181–189 (2009)
- [17] Rodgers, J.R., Cebon, D.: Materials informatics. *MRS Bulletin* 31, 975–980 (2006)
- [18] Rodgers, J.R.: Materials informatics: Knowledge acquisition for materials design. *Abstracts Of Papers Of The American Chemical Society* 226, 302–303 (2003)
- [19] Rajan, K.: Combinatorial materials sciences: Experimental strategies for accelerated knowledge discovery. *Annual Review Of Materials Research* 38, 299–322 (2008)
- [20] Rajan, K.: Combinatorial materials sciences: Experimental strategies for accelerated knowledge discovery. *Annual Review Of Materials Research* 38, 299–322 (2008)
- [21] Moliner, M., Serra, J.M., Corma, A., Argente, E., Valero, S., Botti, V.: Application of artificial neural networks to high-throughput synthesis of zeolites. *Microporous and Mesoporous Materials* 78, 73–81 (2005)
- [22] Fischer, C., Tibbetts, K.J., Morgan, D., Ceder, G.: Predicting crystal structure by merging data mining with quantum mechanics. *Nature Materials* 5, 641–646 (2006)
- [23] Song, Q.G.: A preliminary investigation on materials informatics. *Chinese Science Bulletin* 49, 210–214 (2004)
- [24] Wei, Q.Y., Peng, X.D., Liu, X.G., Xie, W.D.: Materials informatics and study on its further development. *Chinese Science Bulletin* 51, 498–504 (2006)
- [25] Hrubciak, R., George, L., Saxena, S.K., Rajan, K.: A Materials Database for Exploring Material Properties. *Journal of Materials* 61, 59–62 (2009)
- [26] Rajan, K.: Informatics and Integrated Computational Materials Engineering: Part II. *JOM* 61, 47–47 (2009)
- [27] Broderick, S., Suh, C., Nowers, J., Vogel, B., Mallapragada, C., Narasimhan, B., Rajan, K.: Informatics for combinatorial materials science. *JOM* 60, 56–59 (2008)
- [28] Takeuchi, I., Lippmaa, M., Matsumoto, Y.: Combinatorial experimentation and materials informatics. *MRS Bulletin* 31, 999–1003 (2006)
- [29] Hunt, W.H.: Materials informatics: Growing from the bio world. *JOM* 58, 88–88 (2006)
- [30] Inmon, W.H.: *Building the Data Warehouse*, 4th edn. John Wiley and Sons, Inc., New York (2007)
- [31] Li, Y.: Predicting materials properties and behaviour using classification and regression trees. *Materials Science And Engineering A-Structural Materials Properties Microstructure And Processing* 433, 261–268 (2006)
- [32] Westbrook, J.H.: Materials Data On The Internet. *Data Science Journal* 2(25), 198–211 (2003)

Mobile Database Cache Replacement Policies: LRU and PRRP

Hariram Chavan¹ and Suneeta Sane²

¹Information Technology, Terna Engineering College, Mumbai University, India

²Computer Technology, V. J. T. I., Mumbai, India

chavan.hari@gmail.com

sssane@vjti.org.in

Abstract. The continued escalation of manageable, wireless enabled devices with immense storage capacity and powerful CPUs are making the wide spread use of mobile databases a reality. Mobile devices are increasingly used for database driven applications such as product inventory tracking, customer relationship management (CRM), sales order entry etc. In some of these applications Location Dependences Data (LDD) are required. The applications which use LDD are called Location Dependent Information Services (LDIS). These applications have changed the way mobile applications access and manage data. Instead of storing data in a central database, data is being moved closer to applications to increase effectiveness and independence. This trend leads to many interesting problems in mobile database research and cache replacement policies. In mobile database system caching is an effective way to improve the performance, query delay and save bandwidth since new query can be partially executed locally. However, variable data sizes, data updates and frequent client disconnections make cache management a challenge. The conventional method of cache replacement is Least Recently Used (LRU) which takes into account only temporal characteristics of data items. This paper takes into account the parameters like minimum access probability, maximum distance and scope invalidation for cache replacement. These parameters are used and an enhanced version of Prioritized Predicted Region based Cache Replacement Policy (PPRRP) is simulated for the temporal and spatial characteristics. The implementation of both the policies has been done using java. The simulations of both policies reveal the outperformance of PRRP over LRU.

Keywords : Mobile Database, Cache replacement policy, LRU, PRRP, LDD, LDIS.

1 Introduction

The fast development of wireless communications systems and advancement in computer hardware technology has led to the seamlessly converged research area called mobile computing. The mobile computing research area includes the effect of mobility on system, users, data and computing. The seamless mobility has opened up new classes of applications and offers access to information anywhere and anytime.

In client-server model in a mobile computing environment, the clients are mobile units (MUs) that have limited local resources such as notebooks, personal digital assistants (PDA) etc. This environment provided a number of new applications that have been inspiring researches on query processing in mobile databases systems (MDB). The MU communicates with data servers through a wireless link, accessing information at anytime and anywhere. The wireless network is susceptible to frequent disconnections, low-quality communication and limited bandwidth.

The class of data or the data whose value is functionally dependent on location (Location \rightarrow Data value) is called Location Dependent Data (LDD) [14]. For example traffic reports or weather information of a city.

In cellular system a MU is free to move around within the entire area of the coverage. Its movement is random and therefore its geographical location is unpredictable. This situation makes it necessary to locate the mobile unit and record its location when client would like to access something from server. Location Dependent Information Services (LDIS) provide users with the ability to access information related to their current location. They include the location as a part of user's context information [15]. For example, a tourist on a trip to a new city can get help from his MU. He can query based on his personal interest the nearest restaurant, nearest railway station, nearest theater, etc, and can get the response on his MU. Using LDIS leads to many challenges inherent to mobile environments. These challenges include limited bandwidth, limited client power and intermittent connectivity.

The most promising solution to deal with these problems is the data caching technique that will store data copies of frequently used data items in the clients MU. Caching is an effective technique to improve data accessibility and to reduce access cost by caching of frequently accessed data item on client side. On the other hand, it is impossible to hold all accessed data items in the cache due to limitations of cache size on MU. Therefore, it is very important to have efficient cache replacement algorithm to locate a proper subset of data items for eviction from cache.

The cache management policy considers two dimensions: consistency and replacement policies. Cache invalidation aims to keep data consistency between the client's cache and the server database. The cache replacement policy determines which data should be removed from the cache when there is no ample space to hold a new data item.

For the traditional cache replacement policy, the access probability is the most important factor normally considered. Most Recently Used - MRU, Least Recent Used - LRU and Least Recent Used at K Time - LRU-K [7] are examples of the replacement policy. For a mobile computing environment these solutions cannot be convenient since they have only temporal characteristics, generate high network traffic and too much power consumption.

Querying the database directly is a common, robust method for accessing databases on the Internet. As high-speed wireless networks continue to proliferate, the limitations of bandwidth asymmetry and disconnection have a diminishing effect on mobile database design. Thus, the bulk of mobile database research continues to consider limited resources, limited energy, frequent disconnections and bandwidth asymmetry as device characteristics. The common assumption is that accessing central databases for every operation can be prohibitively expensive and can significantly impact that response time and throughput of a mobile application. Caching data and then performing local

queries on the cache is used in many cases. Caching increases the performance of local queries, but comes at a cost of complicated cache maintenance strategies.

This paper deals with describing and implementing PPRRP and compares its result with the standard method. The structure of the paper is as follows: section 2 describes Related Work, section 3 describes Mobile Database System framework. Section 4 taxonomy for cache replacement, section 5 gives performance evaluation and section 6 concludes the paper.

2 Related Work

LRU-K [7] method is an approach to database disk buffering. The basic idea of LRU-K is to keep track of the times of the last K references to popular database pages. This historical information is used to statistically estimate and to discriminate pages that should be kept in buffer. The LRU-K takes into account the temporal characteristics of data access.

The Furthest Away Replacement (FAR) [5] depends on the current location and the direction of the client movement. FAR organizes data items in two sets : In-direction and out-direction. The selection of victim for replacement is based on the current location of user. The assumption for replacement is the locations which are not in the moving direction and furthest from the user will not be visited in near future. The limitation of FAR is it won't consider the access patterns of client and is not very useful for random movement of client.

Probability Area Inverse Distance (PAID) [4] considers both spatial and temporal properties of data for replacement. The cost function for replacement considers the area of valid scope, data distance from current client location to the valid scope and the access probability of client data. However, PAID relies only on the current client movement direction and does not take into account updates to data items.

Prioritized Predicted Region based Cache Replacement Policy (PPRRP) [2] predict valid scope area for client's current position, assigns priority to the cached data items, calculates the cost on the basis of access probability, data size in cache and data distance for predicated region.

Data management in mobile computing environments is especially challenging for the need to process information on the move, to cope with resource limitation and to deal with heterogeneity. Among the applications of mobile data management, LDIS have been identified as one of the most promising area in research and development [12]. The problem has also been studied under various terms such as location-aware, context-aware, or adaptive information systems. Many of the previous work on LDIS treated location as an additional attribute of the data tables. LDIS queries can be processed like ordinary queries except with additional constraints on the location attribute. Caching techniques specially tailored for LDIS or mobile computing environments in general have also been a major research area. Semantic caching techniques employed semantic descriptions of cached items to facilitate better cache admission and replacement decisions that are responsive to the user movement.

3 Mobile Database System Model

We assume a cellular mobile network similar to that Mobile Network Architecture Fig. 1. It consists of mobile client (MC) containing data centric applications roam between wireless cells and accesses a centralized database (fixed host). Some of the fixed hosts, called mobile support stations (MSSs), are augmented with wireless interfaces. The wireless channel is separated into two sub channels: an uplink channel and a downlink channel. The uplink channel is used by MCs to submit queries, while the downlink channel is used by MSSs to answers from the server to target mobile client.

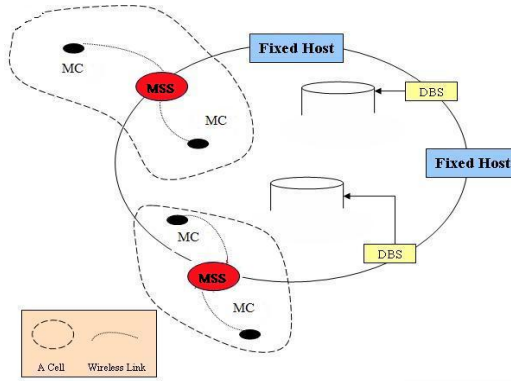


Fig. 1. Mobile Network Architecture

4 Taxonomy for Cache Replacement

In mobile database the taxonomy [13] for cache replacement policy is as shown in Fig. 2.

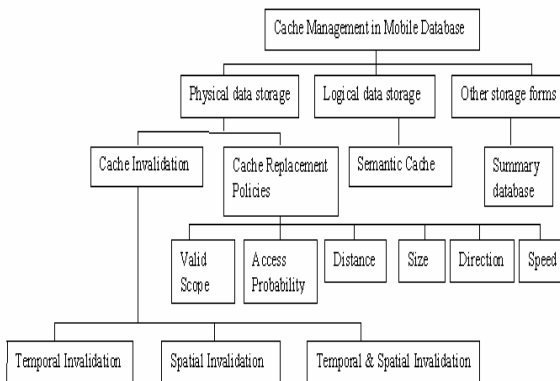


Fig. 2. Taxonomy for cache replacement policies in mobile database

We initiate our classification considering the client cache contents. This will lead to three main categories : physical, logical and other storage forms.

In the physical data storage model the MU cache contents are copies (records or page) of data items from the server. The attempt in this paper is to stick to the physical data storage rather than logical data storage.

The cache invalidation strategies are: Temporal, spatial and temporal & spatial. The Temporal invalidation considers many parameters such as architecture, server type, invalidation method, handoff etc. For example in case of handoff the cache can be kept or completely invalidated.

Spatial invalidation crops up when data values stored in cache become invalid because of the client moves to a new location area (region). The maintenance of a valid cache when the clients move is called location-dependent cache invalidation and a data item can have different value depending on its location [4].

The main feature considered in the spatial invalidation is the valid scope (or valid area). The valid scope of an item value is defined as the set of cells in which the item value is valid. There are different forms to relate the data value with its valid area.

The system set up for this paper deals with following assumptions: Let the space under consideration be divided in physical subspaces as

$$S = \{L_1, L_2, \dots, L_N\}$$

There are data items with

$$D = \{D_1, D_2, \dots, D_M\}$$

Such that each data item is associated with valid scope which is either a set of one or more subspaces from S. Formally shown as

$$D_k = \{L_{k1}, L_{k2}\} \text{ where } L_{k1}, L_{k2} \text{ belong to } S$$

Fig. 3 show the representation of valid scope and Table 1 gives data instance for the same.

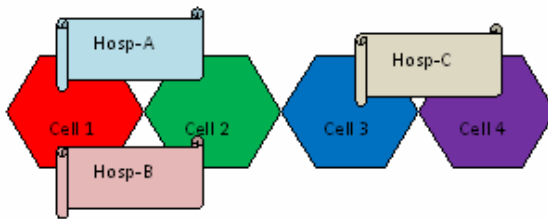


Fig. 3. Valid scope representation for Hospital

Table 1. Valid scope for Hospital data instance

Data instance	Valid Scope
{ nearby-Hosp , {A,B} }	{ 1 , 2 }
{ nearby-Hosp , {C} }	{ 3 , 4 }

The advantage is the complete knowledge of the valid scopes.

The MDS is developed with the following data attributes - valid scope, access probability and distance related to respective location. This centralized database is percolated to MSS in order to make sure that the data relevant in the valid scope is available in the nearest MSS thus making sure that the concept of Prioritized Region is taken into account while considering the cache replacement policy. We have valid scope as an attribute for the location to reduce computation and save the bandwidth. The search for the actual data is done in order as – first select the data as per region (location), get its valid scope and then data with higher access probability will be selected for access so that most probable data item is fetched to the cache. The access probability is updated in database with each access to the cached data item.

When cache does not enough space to store queried data item then, space is created by replacing existing data items from cache based on minimum access probability, maximum distance and scope invalidation. If data items have same valid scope then replacement decision is based on minimum access probability. If valid scope and access probability is same then replacement decision is based on maximum data distance. In some cases, data size plays an important role in replacement. If fetched data item size is large enough and requires replacing more than two data items from cache then replacement is based on maximum equivalent size with minimum access probability, maximum distance and scope invalidation.

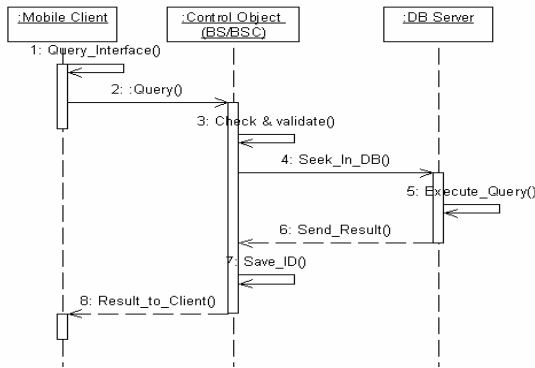


Fig. 4. Sequence diagram for Information Request

The selection of appropriate data item and fetching is done at server side to reduce the computation and save power of client. Fig. 4. shows a scenario of the request for information such as food, ATM, movies, petrol, and hospital etc. The client sends request for information. The server validates the client. For authenticated clients Database Server (DBS) responds with requested information.

5 Performance Evaluation

For implementation, database is created with different regions with locations, location specific resources such as Hospital, Restaurant, ATM, Movies, Blood Bank, Police, Fire Station, Medical 24x7 and resources with different speciality such as Child and Maternity for hospital and as applicable for each resource.

For our evaluation, the results are obtained when the system has reached the stable state, i.e., the client has issued at least 1000 queries, so that the warm-up effect of the client cache is eliminated. We have conducted experiments by varying the query interval and cache size. Query interval is the time interval between two consecutive client queries. In this set of experiments, we vary the query interval from 10 seconds to 200 seconds.

Fig. 5. show the performance of enhanced LRU and PRRRP. To enhance the performance of LRU we have valid scope as added criteria. To enhance the performance of PRRRP we have critical stage-size as additional criteria. The performance is evaluated with respect to change in mean Query Interval. We observe that the performance of PRRRP is far better than LRU. As the query interval increases, cache hit ratio decreases, because the client would make more movements between two successive queries, thus has low probability to remain in the same valid scope queried previously when a new query is issued.

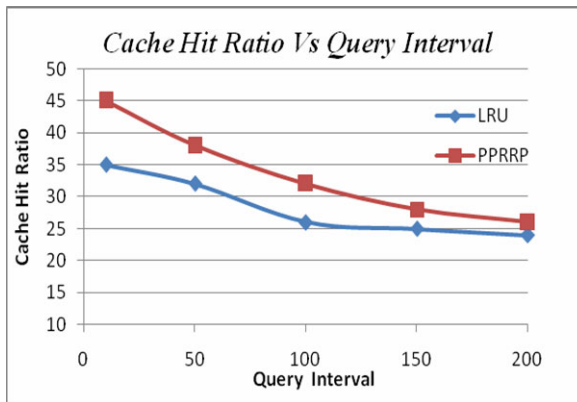


Fig. 5. Cache hit Ratio vs Query Interval

Fig. 6. depicts the effect of cache size on performance of LRU and PRRRP replacement policies. The performance of PRRRP will be increased substantially with increase in cache size so this result could be used to decide the optimal cache size in the mobile client.

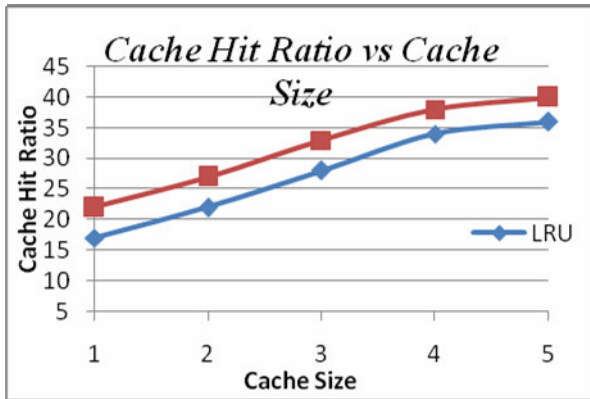


Fig. 6. Cache hit Ratio vs Cache Size

6 Conclusion

In this paper, we have presented a performance evaluation of LRU and PRRP cache replacement policies. The LRU is temporal while PRRP takes into account both the spatial and temporal properties of client movement and access patterns to improve caching performance. Simulation results for query interval and cache size show that the PRRP has significant improvement in the performance than the LRU. In our future work we would like to incorporate Furthest Away Replacement (FAR) as part of comparison with PRRP and our own Markov Model based cache replacement policy for prediction and replacement.

References

- [1] Barbara, D.: Mobile Computing and Databases: A Survey. Proc. of IEEE Trans. on Knowledge and Data Engg. 11(1) (1999)
- [2] Kumar, A., Misra, M., Sarje, A.K.: A Predicated Region based Cache Replacement Policy for Location Dependent Data In Mobile Environment. IEEE, Los Alamitos (2006)
- [3] Oliver, E.: A Survey of Mobile Database Caching Strategies. David R. Cheriton School of Computer Science University of Waterloo
- [4] Zheng, B., Xu, J., Lee, D.L.: Cache Invalidation and Replacement Strategies for Location-Dependent Data in Mobile Environments. Proc. Of IEEE Trans. on Comp. 51(10) (2002)
- [5] Ren, Q., Dhunham, M.H.: Using Semantic Caching to Manage Location Dependent Data in Mobile Computing. In: Proc. of ACMIEEE MobiCom, pp. 210–221 (2000)
- [6] Dar, S., Franklin, M.J., Jonsson, B., Srivastava, D., Tan, M.: Semantic Data Caching and Replacement. In: Proceedings of the 22nd VLDB Conference Mumbai, Bombay, India (1996)
- [7] O’Neil, E., O’Neil, P.: The LRU-k page replacement algorithm for database disk buffering. In: Proc. of the ACM SIGMOD, pp. 296–306 (1993)

- [8] Fife, L.D., Gruenwald, L.: Research Issues for Data Communication in Mobile Ad-Hoc Network Database Systems. Brigham Young University – Hawaii, Computer Science Department, Laie, HI 96762
- [9] Xu, J., Hu, Q., Lee, W.-C., Lee, D.L.: Performance Evaluation of an Optimal Cache Replacement Policy for Wireless Data Dissemination. *IEEE Transactions On Knowledge And Data Engineering* 16(1) (January 2004)
- [10] Zhang, J., Gruenwald, L.: Spatial and Temporal Aware, Trajectory Mobility Profile Based Location Management for Mobile Computing, [HREFI], <http://students.ou.edu/Z/Jianting.Zhang-1/Html/STAMobility.htm>
- [11] Zebchuk, J., Makineni, S., Newell, D.: Re-Examining Cache Replacement Policies. *IEEE, Los Alamitos* (2008) 978-1-4244-2658-4/08/\$25.00 ©2008
- [12] Wu, S.-Y., Wu, K.-T.: Dynamic Data Management for Location Based Services in Mobile Environments. In: *Proceedings of the Seventh International Database Engineering and Applications Symposium (IDEAS 2003)*, pp. 1098–8068. *IEEE, Los Alamitos* (2003) 1098-8068/03 \$17.00 ©2003
- [13] Mânica, H., de Camargo, M.S.: Alternatives for Ccache Management in Mobile Computing. In: *IADIS International Conference Applied Computing* (2004)
- [14] Dunham, M.H., Kumar, V.: Dependent Data and its Management in Mobile Databases. In: *9th International Workshop on Database and Expert Systems Applications (DEXA 1998)*, Vienna, Austria, August 26-28, p. 414 (1998)
- [15] Hiary, H., Mishael, Q., Al-Sharaeh, S.: Investigating Cache Technique for Location of Dependent Information Services in Mobile Environments. *European Journal of Scientific Research* 38(2), 172–179 (2009)

A Novel Data Mining Approach for Performance Improvement of EBGM Based Face Recognition Engine to Handle Large Database

Soma Mitra, Suparna Parua, Apurba Das,
and Debasis Mazumdar

CDAC, Kolkata, Salt Lake Electronics Complex, Kolkata, India
{soma.mitra,suparna.parua,apurba.das,debasis.mazumdar}@cdackolkata.in

Abstract. Computerized human face recognition is a complex task of deformable pattern recognition. The principal source of complexities lies in the significant inter-class overlapping of faces due to the variations caused by different poses, illuminations, and expressions (PIE). Elastic Bunch Graph Matching (EBGM) is a feature-based face recognition algorithm which has been used fairly reliably to determine facial attributes from an image. It extracts the texture using Gabor wavelets around a set of biometric landmark points on a face, and generates a level graph. One of the degrading factor of the performance of EBGM based face recognition system is the size of the database, particularly when the database size is in tuned to millions, the performance of FRE falls drastically. In the present paper data mining approach is presented to improve the performance of the EBGM based face recognition engine in case of large database. We have proposed entropy based decision tree for feature selection and feature hierarchy. The selected features are taken to form suitable feature vector for Fuzzy C-means clustering. The clustered set becomes the reduced search space for the query face. Improvement in the performance of the EBGM based FRE is presented with suitable experimental results.

Keywords: Face recognition, Gabor jets, Entropy, Decision tree, Fuzzy C-means clustering, Clustering validity index, EBGM.

1 Introduction

Recognition of human face is a complex visual pattern recognition problem. The complexities of the problem are mainly due to significant intra-class variations in appearance pertaining to pose, facial expressions, ageing, illumination, and ornamental variations. The earliest research on automatic machine recognition of faces can be traced back to the 1970s in the engineering literature [1]. In the Elastic Bunch Graph matching algorithm (EBGM), proposed by Wiskott et al. [2], they have determined facial attributes from image in an unique way. EBGM can even recognize human faces from a large database consisting of only one image per person. The method of face recognition using EBGM and Gabor

wavelet has been discussed in section two. In the present paper, decision tree based data mining is used to handle large image database. Instead of searching for match into the entire database of D number of face images, if we can cluster the entire database into N number of classes, the query face needs to search for the match into approximately D/N number of face images only. With this objective, we have determined the most discernable features for face recognition using decision tree in section three. In the fourth section we have proposed the choice of parameters for Fuzzy C-means clustering to be applied into the large face database. At the end, we have presented the rank histogram of the EBGM-Gabor based face recognition engine to show the improvement of the face recognition engine due to data mining through decision tree and Fuzzy C-means clustering.

2 Elastic Bunch Graph Matching and the Face Manifolds

In tasks like face recognition, involving within-class discrimination of objects, it is necessary to have information specific to the structure common in all objects of the class. The bunch graphs are stacks of a moderate number of different faces, jet-sampled to an appropriate set of fiducial points (placed over eyes, nasal bridge, tip of the nose, mouth corner, etc.). It should be noted here that the original EBGM algorithm was proposed [2] to recognize facial images of frontal views. In the real life scenario, the recognition across pose variation is essential. In EBGM, two data structures are used to pursue two different tasks. The first data structure, named as face bunch graph (FBG) includes a wide range of possible variations in the appearance of faces, such as differently shaped eyes, noses, mouth, different types of beards, variations due to sex, age etc. The face bunch graph combines a representative set of different possible variations around the fiducial points into a stack like structure. Another data structure used in EBGM is technically called the image graph, which represent a face in the form of a graph. In image graph, each node corresponds to fiducial point and at each node a stack of Gabor jet is stored to represent textural information at different orientations and frequencies. For face recognition across pose variation one needs to develop different bunch graphs for different poses, which is not at all practical solution in the real life scenario. In our face recognition engine, we use a single bunch graph for the frontal view and selection of the location of the fiducial point at non-frontal poses are made using separate sets of feature extraction algorithm. In the next section we describe the feature selection algorithm and its validation.

3 Feature Selection and Entropy Based Feature Hierarchy

To improve the performance of the face recognition system in terms of reduction of response time and rank, we need to select the most important features from the feature vector of 560 points. After getting the feature hierarchy, we can take the most important feature to cluster the entire face gallery, offline. Then the EBGM and Gabor based face recognition engine, discussed in the previous section will

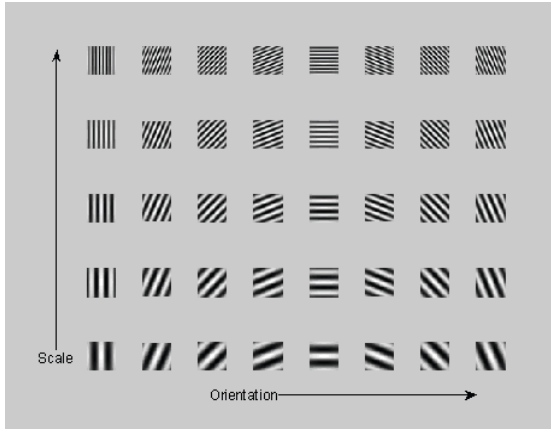


Fig. 1. Schematic representation of the Gabor filter banks in 5 scales and 8 orientations

run into the reduced set of face database. In the present section, feature selection from the 560 point long feature vector is done and then a feature hierarchy is obtained using entropy based decision tree [4]. The top most important features will be considered for clustering as discussed in the next section.

3.1 Selection of Optimized Set of Features

In each face 14 fiducial points are located. At each feature point 40 Gabor filters (at 5 scales and 8 orientations), as shown in Fig. 1 are convolved to extract the feature vector of dimension forty. Hence, to represent a single face a feature vector of dimension 560 (14×40) is computed. To select important feature points, we have taken facial images of 5 different subjects having 10 different images each (varied in PIE). Therefore, for each feature, we can have a 50×40 matrix. Each 50×40 matrix results in 40×40 covariance matrix. The diagonal items of each covariance matrix are essentially the variance of similar sub-features. Next, the diagonal vector is sorted to get the first 3 values which gives maximally varied 3 feature points amongst 40 feature points of each fiducial point. The features which have been selected by this process are listed below.

A close scrutiny of the result shows that the selected features can be ranked in descending order, based on their discernibility capacity. The result is quite in agreement with human perception. If we look at the indexes of eyebrow, index 7 means 2nd orientation 2nd scale as 5 scales or frequencies of Gabor wavelet are there per orientation, index 12 means 3rd orientation and 2nd scale, index 2 means 1st orientation of 2nd scale. Therefore, it can be inferred that, for eyebrow, 2nd scale is important. All the Gabor jets in that particular orientation are responding. For the other feature points also, the pattern is shown in Fig. 5.

Table 1. Maximally varied sub-features of each feature point

Sl. No.	Features	Maximally varied features		
		1st	2nd	3rd
1	Eyebrow features	B ₇	B ₁₂	B ₂
2	Eye features	E ₃₀	E ₂₅	E ₂₀
3	Lip features	L ₃₆	L ₃₁	L ₂₆
4	Nose features	N ₆	N ₂₁	N ₃₅

Table 2. Self-similarity measure by omission of individual feature point

Sl. No.	Omission of feature point	Similarity measure with same face
1	Eyebrow	0.914360
2	Eye	0.923622
3	Lip	0.927570
4	Nose	0.928571

3.2 Computation of Feature Hierarchy

To form a hierarchy into the selected features from the previous section it is needed to form feature vector for the ease of clustering. From the 14 fiducial points, we need to find out the most important features with their ranks and importance in the entire face recognition activity. Therefore, using different algorithms the features are arranged in hierarchy as discussed in this section and the features from the top of the hierarchy are used for clustering as discussed in the next section.

Computation of feature hierarchy using Entropy Based Decision Tree.

A decision tree is a popular classification method that results in flow-chart like tree structure where each node denotes a test on an attribute value and each branch represents an outcome of the test. The tree leaves represents the classes. Decision tree algorithm is a relatively simple top-down greedy algorithm. The aim of the algorithm is to build a tree that has leaves that are as homogeneous as possible [3]. The major step of the algorithm is to continue to divide leaves that are not homogeneous into leaves that are as homogeneous as possible until no further division is possible. The discriminatory power of each attribute may be evaluated in a number of different ways. Here entropy based decision tree has been built to get the hierarchy of the selected features from the previous sub-section. C4.5 algorithm of Decision tree is applied here [4]. Here, in this experiment, each face is designated by 12 feature vales. Feature is formed using 3 maximally varied features (as discussed in the previous section) from 4 fiducial points as left eyebrow, left outer eye corner, left lip corner and nose tip. The right eye-brow, eye, lips are not taken for symmetry. Ten subjects with 5 poses each (frontal, 5 degree left, 5 degree right, 10 degree left, 10 degree right) are taken, as shown in Fig 2. As shown in the Fig 3, the largest information gain



Fig. 2. One subject with 5 poses: Frontal, 5L, 5R, 10L, 10R (Left to right)

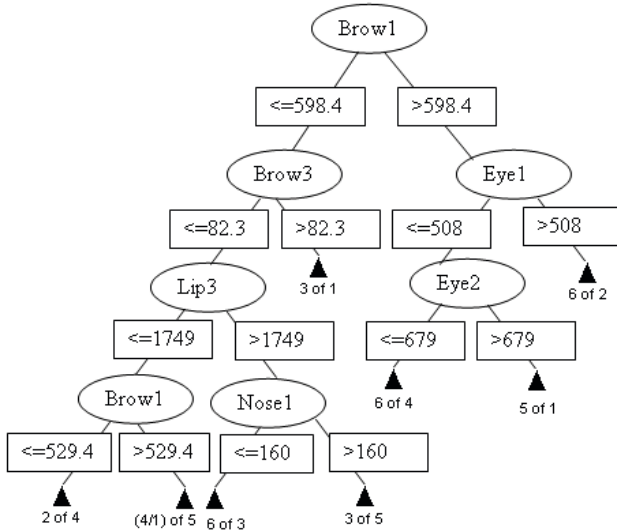


Fig. 3. Entropy based decision tree showing feature hierarchy

in terms of entropy difference is provided by the attribute Brow1 (1st selected feature point in the previous section) and that is responsible for largest split between fairly homogeneous classes. In the next level also Brow3 comes along with Eye1. Ultimately, we have got different non-overlapping classes each with homogeneous faces and surely with faces of same subjects with different poses. From this hierarchy, it is clear that, in the time of clustering, eyebrow and eye will have maximum emphasis.

Computation of feature hierarchy by selective omission of feature points. In the next experiment, we have artificially omitted all of the feature points one by one and tested the similarity score by matching with other features. The faces with omitted features are shown in the Fig. 4. In this experiment also the similarity scores presented are as shown in Table 2.

It is clearly seen that, the similarity score is maximally hampered due to absence of eyebrow feature. The reduction in similarity score due to absence of feature at eye corner, lip corner and nose tip comes next, respectively. Therefore, the feature hierarchy is the same as obtained from the entropy based decision tree as: eyebrow to eye corner to lip corner to nose tip. The result obtained by the decision tree is validated.



Fig. 4. Faces with artificial occluded fiducial points

4 Fuzzy C-means Clustering for Performance Improvement of the FRE

In most of the practical classification problems, overlapping of classes occur due to the uncertainty inherently present in any natural data. Practically, a query point may belong to different classes with different membership values ranging between 0 and 1. In our problem, classes consisting of different facial images of single subjects are also overlapping in nature. We have applied the Fuzzy C-means clustering algorithm [5] in six dimensional feature space. According to the selected features from the previous section, variances of the Gabor filtered output of six particular fiducial points, as left and right eyebrow (fiducial point number 1 and 2 respectively), inner (fiducial point number 5 and 8) and outer (fiducial point number 3 and 10) eye corners of two eyes are considered. Variations of three features for some members of each cluster are shown in the Fig. 5. The other features are symmetrical. The apparent similarity of the feature vectors as shown in Fig. 5 inspire the human perception conclude that they belong to the same class. Variance of each feature vectors are computed and considered as the feature to design the discriminator. The Fuzzy C-means clustering algorithm is run on the database of 1000 facial images with the C value 3, 4, 5, 6 and 7 respectively. The D. B. (Devis-Bouldin) index [6] have been calculated in each case as shown in Table 3. The minimum value of D. B. index is obtained in case of $C = 3$. However, in the obtained cluster we found that substantial amount of overlapping exist between two clusters. Hence in case of query, search is required for both the overlapping classes. For example, in Table-3 ($C=3$ case) cluster 2 and cluster 3 are overlapping and search is required to be for $328+176=504$ images which is almost half of the total database size (1000). Instead, if we choose the second least D. B. index, we need to divide the entire database into four clusters (i.e., $C = 4$) and overlapping is observed within class 2 and 4, where one query needs to search $141+219=360$ instead of 1000. This decrease in search space by an amount of 70%, for that particular query which falls under class 2 or 4 as shown in Fig. 6. Therefore, we have chosen the 2nd least D. B. which divides the entire database into four classes.

Table 3. Number of members in each class with respect to D.B. index

No. of clusters	Number of members in							D. B. Index
	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	
7	124	144	146	98	82	291	115	7.0577
6	105	189	119	303	144	141	-----	4.7195
5	228	166	151	126	329	-----	-----	4.3181
4	408	141	232	219	-----	-----	-----	3.06
3	496	328	176	-----	-----	-----	-----	2.9

4.1 Reducing the Search Space for FRE Using Fuzzy C-Means

In our problem of face recognition from a large gallery, the concept of overlapping classes needs to be considered. Therefore Fuzzy C-means clustering algorithm is nicely applicable. The query face may belong to different classes with different membership. To reduce the FAR and FRR of the system, Fuzzy C-means clustering is employed instead of hard clustering like K-means [7]. In our case, we have taken the most important fiducial points responsible for discrimination of faces of different human being. Total six fiducial points are selected as left and right eyebrow, inner and outer eye corners of two eyes. As, the variation of the vector is the key parameter for facial image matching, we have chosen statistical variance of each of the fiducial points. So each face is now designated with a new vector of length 1×6 only. Therefore, input of the Fuzzy C-means clustering algorithm should be a two dimensional vector of size N×6, where N is the size of the database. Now these N images are divided in 4 different clusters with respective membership values. This clustering is done off-line and just after enrolment of gallery images. Whenever the administrator requires modifying the gallery with addition or deletion of images, entire process of clustering needs to be employed again. In the time of matching, first the 6 particular feature vectors are extracted from the query face and the distance is calculated individually between the 1×6 dimensional vector and the 4 pre-clustered centers. Depending on the least distances, the memberships are assigned to the query as $\mu_{Cluster1}(Query)$, $\mu_{Cluster2}(Query)$, $\mu_{Cluster3}(Query)$ and $\mu_{Cluster4}(Query)$. The number of clusters is determined from experiments done on a large database of images. If we restrict the number of classes to 4, the Devis-Bouldin index [6] is minimized, i.e., ratio of intra class to inter-class difference is minimized. The search space is now deduced for that particular query. The maximum membership cluster only will be searched. Firstly, the entire image gallery of 1000 facial images is clustered into 4 classes, offline. The six dimensional feature vector is formed by the variances of the Gabor filtered output of six particular fiducial points. From feature hierarchy as dictated by the decision tree, the 6 fiducial points are taken as left eyebrow (feature id 1), right eyebrow (feature id 2), left eye outer corner (feature id 3), left eye inner corner (feature id 5), right eye inner corner (feature id 8) and right eye outer corner (feature id 10). When a query face comes, the six-dimensional feature vector is derived from the automatically localized fiducial points after Gabor filtering followed by taking the variances.

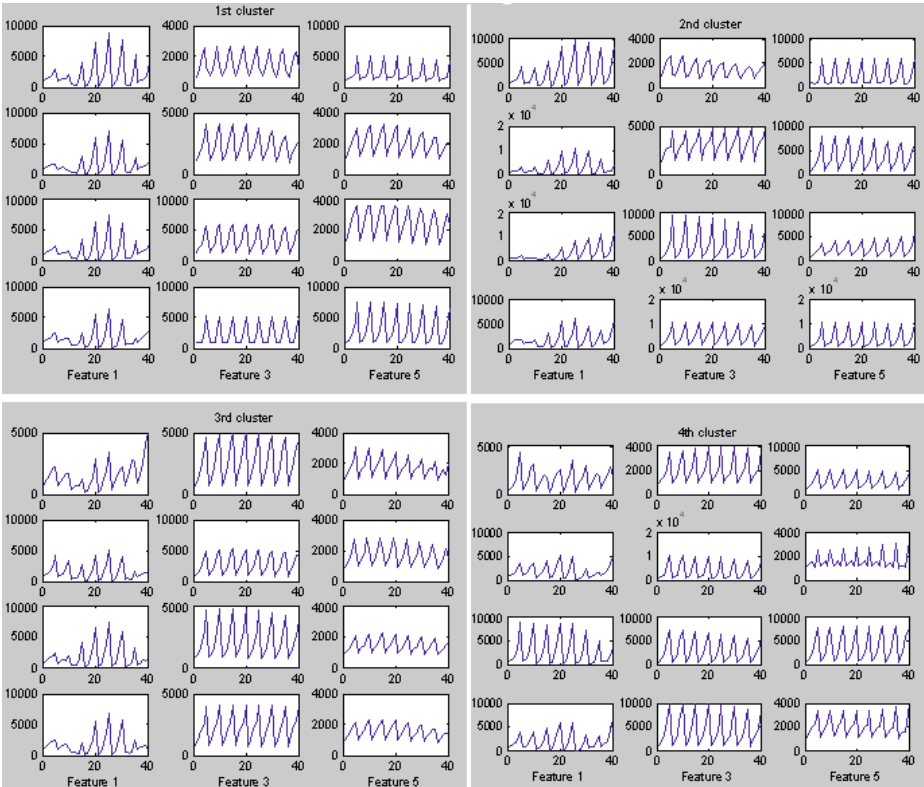


Fig. 5. Variations of three features for some members of each cluster

Next, distances from all the C cluster centers are calculated. The minimum distance cluster and second minimum distance clusters are marked as ζ_1 and ζ_2 respectively. If the difference between the fuzzy membership of the query face Q belonging to cluster ζ_1 i.e., $\mu_{\zeta_1}(Q)$ and membership of the query face belonging to cluster ζ_2 i.e., $\mu_{\zeta_2}(Q)$ is more than 0.2, then face image matching (image identification) will be carried on upon cluster $\mu_{\zeta_1}(Q)$ instead of total gallery. On the other hand, if the difference $\mu_{\zeta_1}(Q) - \mu_{\zeta_2}(Q)$ is less than 0.2, both the clusters ζ_1 and ζ_2 will be incorporated into the search space. Then also the search space would be reduced by at least 60%. Face matching is done taking dot product between the Gabor filtered feature vectors (length=560) of the query and each member of the reduced face database, followed by choosing the highest values of similarity. As the search space is reduced, obviously the mean rank of face recognition engine will be improved as shown in the Fig 7 and also, the speed of the engine will be increased by a significant amount, i.e., the average response time is decreased from 55 seconds to 15 seconds for a database having 1000 facial images.

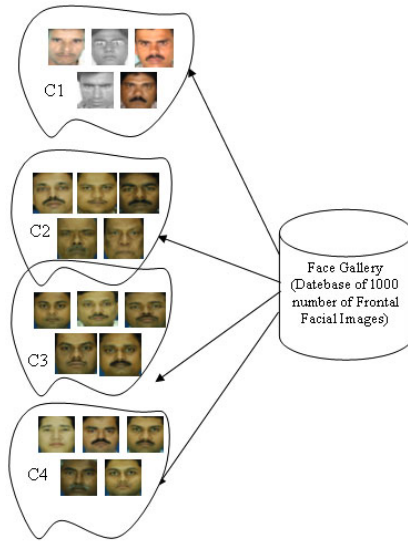


Fig. 6. Clustering of face images using Fuzzy C-means clustering

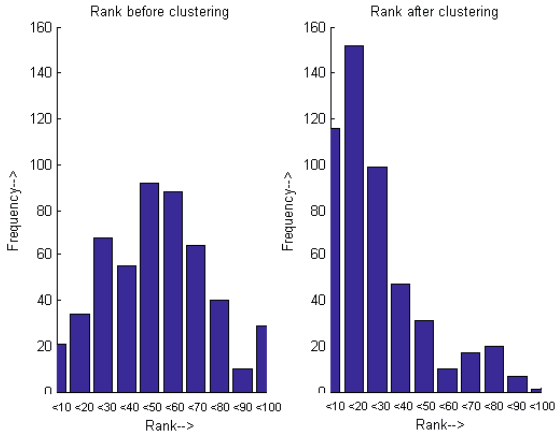


Fig. 7. Rank histogram of the face recognition engine before after Fuzzy C-means clustering on six dimensional feature space

5 Conclusion

In this present paper we have derived the decision tree based feature hierarchy for optimized feature selection. On the basis of the selected features, we have applied Fuzzy C-means clustering for improvement of the performance of the EBGm-Gabor based face recognition engine. We have tested our result on CDAC

face database. According to the experimental results, we have shown that the recognition speed has been increased as well as the mean rank of the query face has been improved. A novel technique of data mining by decision tree and Fuzzy C-means clustering is proposed for improvement of the performance of face recognition system.

Acknowledgement

The authors gratefully acknowledge the financial support of DIT, MCIT, Government of India.

References

1. Kanade, T.: Computer recognition of human faces. *Interdisciplinary Systems Research* 47 (1977)
2. Wiskott, L., Fellous, J.M., Kruger, N., Malsburg, C.V.D.: Intelligent Biometric Techniques in Fingerprint and Face Recognition, Face Recognition in Elastic Bunch Graph matching. *International Series on Computational Intelligence*, pp. 357–396. CRC Press, Boca Raton (1999)
3. Gupta, G.K.: Introduction to data mining and case studies. Prentice Hall of India Pvt. Ltd., Englewood Cliffs (2006)
4. Quinlan, J.R.: C4.5 programs for machine learning. Morgan Kaufmann, San Francisco (1992)
5. Bezdec, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981)
6. Tou, J.T., Gonzales, R.C.: Pattern Recognition Principles. Addison-Wesley, Massachusetts (1974)
7. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice Hall Advance Reference Series: Computer Science. Prentice Hall College Div. (1988)

Efficient Density Based Outlier Handling Technique in Data Mining

Krishna Gopal Sharma¹, Anant Ram¹, and Yashpal Singh²

¹ Department of Computer Science, G.L.A Institute of Technology and Management,
Mathura, India

² Department of Computer Science, Bundelkhand Institute of Engineering and Technology,
Jhansi, India

hollyhoc@rediffmail.com, anantram08@gmail.com,
yash_biet@yahoo.co.in

Abstract. Local Outlier Factor (LOF) is an important and well known density based outlier handling algorithm, which quantifies, how much, an object is outlying, in a given database. In this paper, first we discuss LOF and its variants (LOF' and LOF'') and then we propose an efficient density based outlier handling algorithm, which is inspired by LOF and LOF'. This algorithm not only focuses on the density-based notion to discover local outliers but also reduces the number of passes to scan the complete database. This method calculates the MinPts-dist variance for every object. If MinPts-dist variance of an object is greater than a specified threshold value than that object is considered as an outlier. The experimental results show that the proposed outlier handling algorithm detects outliers more effectively.

Keywords: Outlier-ness, lrd, MinPts-Neighborhood, MinPts-dist, MinPts-dist Variance.

1 Introduction

Outliers are important because as pointed out in [8], "One person's noise could be another person's signal". If outlier is a noise, then it could degrade the predictive accuracy of data mining algorithms. So its detection is important, for its removal, to improve the accuracy of prediction [9], and if it contains the valuable knowledge hidden then it's even more important to find it out. Definition of outlier, given by Hawkin, states that an outlier is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism[3]. Detection of outliers is useful in finding criminal activities in E-commerce, telecom and credit card frauds, loan approval, intrusion detection, video surveillance, pharmaceutical research, transportation, public health, public safety etc [1][2]. There are a number of outlier detection approaches, out of which density based approach is more important. Density based approach provides the concept of local outlier. Local outliers have significant density difference with their neighbors. They reside in low density area.

In this paper, we emphasize that there exist many density-based local outlier detection techniques which are robust in run time complexity but still are not able to detect

the good quantity of objects, which are outliers in the database. We propose a density based algorithm MDV (MinPts-dist variance) which is inspired by Local Outlier Factor (LOF) [4] and LOF' [1]. Its run time complexity is not greater than that of LOF; also the no. of passes to scan the complete database is less. This algorithm gives better result than LOF and its variants (LOF' and LOF''). The remainder of the paper is organized as follows: In section 2, some related work in outlier detection is discussed. In section 3, the proposed enhanced algorithm is explained in detail. In section 4, Complexity analysis is discussed. In section 5 we present the experimental results. Finally, Section 6 concludes the paper.

2 Related Work

A. LOF [4]

LOF (Local Outlier Factor) [4], introduced by Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng and Jörg Sander, is the first concept that quantifies how outlying an object is relative to its neighboring objects in the database and the LOF value of the object is the average of the ratios of local reachability densities of neighbors divided by local reachability density of the object itself. The cardinality of the neighborhood of the object is the no. of objects residing in the area determined by minimum number of objects (MinPts). MinPts is a user supplied parameter. Actually, LOF borrowed the concept such as MinPts and reachability distance from DBSCAN [5] and OPTICS [6]. Terms and concepts necessary to explain LOF algorithm can be defined as follows.

For all the definitions given below, k is taken as any positive number.

Definition 1: (k -distance of an object p) The k -distance of an object p , denoted as k -distance (p), is the distance between object p and its k th nearest neighbor.

Definition 2: (k -distance neighborhood of an object p) Given the k -distance of p , the k -distance neighborhood of p contains every object whose distance from p is not greater than the k -distance, i.e. All objects that are residing inside or on the circumference of the circle with centre p and radius k -distance(p). We will use $N_k(p)$ to denote k -distance neighborhood of p .

Definition 3: (reachability distance of an object p with respect to object o) The reachability distance of object p with respect to object o is the maximum out of k -distance of an object o and real distance between p and o . We denote this by $\text{reach-dist}_k(p, o)$.

Example: - Consider Fig. 1 for explanation. Values of k taken here is 4. So 4-distance of object p in Fig-1 is the distance between p and q because q is the 4th nearest neighbor of p . The 4-distance neighborhood of p (the value of $N_4(p)$) is 6 as only 6 objects are lying inside or on the periphery of circle with centre p and radius k -distance(p). Now, reachability distance of r with respect to p is k -distance(p) i.e. distance between p and q while reachability distance of s with respect to p is actual distance between p and s as described in definition 3.

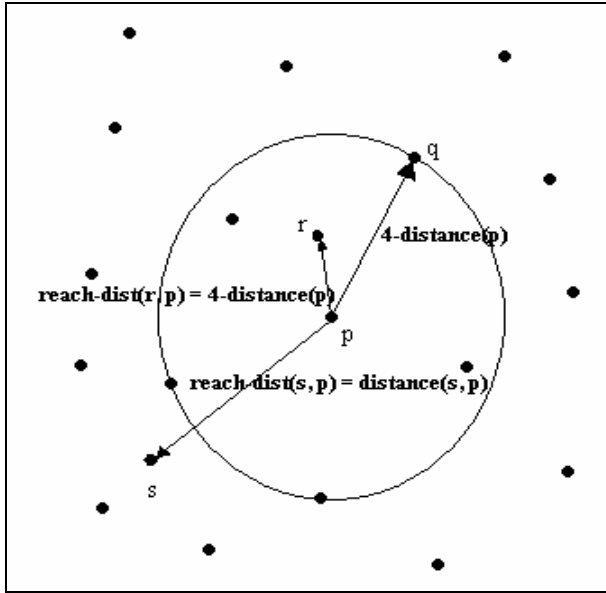


Fig. 1. Depiction of k -distance, k -distance neighborhood, reachability distance for $k = 4$

Let MinPts be the user supplied value for k and ρ be the indication of mass and r be the value for reachability distance of object p with respect to object o , $\text{reach-dist}_{\text{MinPts}}(p, o)$ for $o \in N_{\text{MinPts}}(p)$, be a measure of the volume to determine the density in the neighborhood of an object p , then

Definition 4: (local reachability density of an object p) The local reachability density (lrd) of p is defined as

$$\text{lrd}_{\text{MinPts}}(p) = 1 / \left[\frac{\sum_{o \in N_{\text{MinPts}}(p)} \text{reach-dist}_{\text{MinPts}}(p, o)}{|N_{\text{MinPts}}(p)|} \right]$$

Clearly, the local reachability density of an object p is one divided by the average reachability distance based on the MinPts -distance neighborhood of p .

Definition 5: (local outlier factor of an object p) The local outlier factor of an object p is defined as

$$\text{LOF}_{\text{MinPts}}(p) = \frac{\sum_{o \in N_{\text{MinPts}}(p)} \frac{\text{lrd}_{\text{MinPts}}(o)}{\text{lrd}_{\text{MinPts}}(p)}}{|N_{\text{MinPts}}(p)|}$$

It is the ratio of the average of local reachability density of p 's MinPts -distance neighborhood to that of object p itself. Formula makes it clear that if lrd of object p is

low and lrd of p 's neighborhood is high than LOF value for object p will be higher indicating its higher outlier-ness. As already mentioned, the outlier factor of an object p captures the degree to which we call p an outlier.

B. LOF' [1]

It is clear from the above discussion of LOF that local reachability density captures the notion of density around a data object. But MinPts-dist already captures this notion: a large MinPts-dist corresponds to a sparse region; a small MinPts-dist corresponds to a dense region [1]. LOF' is defined below.

Definition 6

$$\text{LOF}'_{\text{Minpts}}(p) = \frac{\sum_{o \in N_{\text{MinPts}}(p)} \frac{\text{MinPts-dist}(p)}{\text{MinPts-dist}(o)}}{|N_{\text{MinPts}}(p)|}$$

As defined above LOF' is the average ratio of MinPts-dist of an object and that of its neighbors within MinPts-dist. One advantage of this is its Simplicity and another advantage is: computation of LOF' becomes more efficient in comparison to LOF because one pass over the data is saved (in LOF' reachability distance and local reachability density are not to be calculated as they are not the part of definition).

C. LOF'' [1]

If we set MinPts too low, than there is a possibility that outliers, close to each other, may form small groups of outlying objects and can be misidentified as small cluster (as MinPts is the minimum number of objects that can be considered as cluster). On the other side, If we set MinPts too high, some outlying objects near the dense cluster may be misidentified as a part of cluster (as MinPts is also used to calculate the density of object).

In fact, there are two different neighborhoods: one for comparing the densities (MinPts1) and other for computing the densities (MinPts2). In LOF, these two neighborhoods are identical but in LOF'' they are not. LOF'' is defined in [1] as:-

Definition 7

$$\text{LOF}''_{\text{MinPts1, MinPts2}}(p) = \frac{\sum_{o \in N_{\text{MinPts1}}(p)} \frac{\text{lrd}_{\text{MinPts2}}(o)}{\text{lrd}_{\text{MinPts2}}(p)}}{|N_{\text{MinPts1}}(p)|}$$

MinPts2 is assigned a value less than MinPts1. LOF'' displays similar property as depicted by LOF and LOF' i.e. objects deep inside a cluster have LOF'' values close to 1[1].

3 Proposed (MDV) Algorithm

From the above discussion, we know that MinPts-dist of an object captures the notion of density around that object. Small MinPts-dist maps to dense region i.e. high density

while large MinPts-dist maps to sparse region (low density). In LOF' local outlier factor is computed by using MinPts-dist in place of lrd and formulation was simple and efficient.

In this section we propose the algorithm that is based on density based notion, it requires equal number of passes as that of LOF' [1] and produces the better results among all. It requires only two passes to complete it.

Pass 1: In this pass, algorithm calculates MinPts-dist for each object which is defined below and it also calculate the MinPts-neighborhood.

Definition 8: (MinPts-distance of an object p) For any positive integer MinPts, the MinPts-distance of an object p, denoted as MinPts-dist(p), is defined as the distance $d(p, o)$ between p and an object $o \in D$ (dataset) such that:

(i) For at least MinPts objects $o' \in D \setminus \{p\}$ it holds that $d(p, o') \leq d(p, o)$, and

(ii) For at most $(\text{MinPts} - 1)$ objects $o' \in D \setminus \{p\}$ it holds that $d(p, o') < d(p, o)$.

Definition 9: (MinPts-distance neighborhood of an object p) Given the MinPts-distance of an object p, the MinPts-distance neighborhood of p contains every object, whose distance from p is not greater than the MinPts-distance, i.e.

$$N_{\text{MinPts}}(p) = \{q \in D \setminus \{p\} \mid d(p, q) \leq \text{MinPts} - \text{dist}(p)\}$$

Here object q is the MinPts-nearest neighbor of p.

Pass 2: In this second pass; algorithm calculates the MinPts-dist variance for every object of the database. It is defined as follows:

Definition 10: (MinPts-dist variance) Variance of a random variable or distribution, as defined in Wikipedia, is a measure of the amount of variation within the values of that variable. It is an indicator of the “spread” of a distribution [10].

Let we consider the variance of MinPts-dist about a data point (object) p with respect to data points that are MinPts-distance neighborhood of p and call this variance as MinPts-dist variance. It is denoted by MDV. We formulated it as follows:-

$$MDV_{\text{MinPts}}(p) = \frac{\sum_{o \in N_{\text{MinPts}}(p)} (\text{MinPts} - \text{dist}(p) - \text{MinPts} - \text{dist}(o))^2}{|N_{\text{MinPts}}(p)|}$$

It is clear from the above formulation that MinPts-dist variance is the average of square of difference between MinPts-dist of p and that of its MinPts-distance neighborhood. Higher value of MDV of an object p i.e. $MDV(p) > \delta$ (Where δ is the threshold value) indicates the greater degree of outlier-ness of the object with respect to its surrounding (MinPts-dist neighborhood).

Three cases arise as shown in fig. 2. Here, p is the object whose outlier-ness is being checked. MinPts value considered here is 4. Circle around the object points are their respective MinPts-distances.

Case 1: In it, density around p is quite low (i.e. p belongs to sparse region as it is clear in fig. 2). The MinPts-distance of p is quite large to the MinPts-distance of data

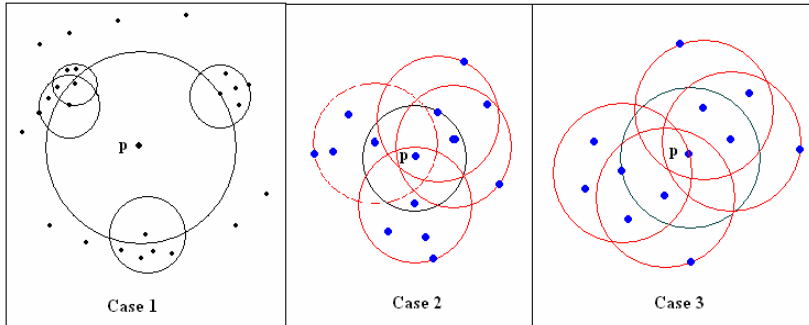


Fig. 2. Three different cases of density variation around p

object that are in MinPts-distance neighborhood of p . Therefore MinPts-dist variance of p is large indicating its outlier-ness.

Case 2: Here, density around p is large, giving suspicion of being its outlier-ness. MinPts-distance of p , for this case, is small to the MinPts-distance of data object that are in MinPts-distance neighborhood of p . Again MDV is large indicating its outlier-ness.

Case 3: In case 3, p and its MinPts-distance neighbors have approximately equal MinPts-distance (i.e. objects are evenly distribute density wise). Therefore, p is not an outlier.

Our algorithm considers case 2, as a possibility of being an object outlier while other outlier detecting algorithms generally do not consider this.

Algorithm 1. MDV Algorithm

MDV Algorithms (D , MinPts, δ)

1. For each object $p \in D$ (Dataset).
2. Calculate the MinPts-dist (p).
3. Calculate the MinPts-distance neighborhood (p).
4. For each object $p \in D$ (Dataset).
5. Calculate MinPts-dist variance (p).
6. If MinPts-dist variance (p) $> \delta$
7. Then object (p) is an outlier.
8. Else object (p) is not an outlier.
9. End

4 Complexity Analysis

Outlier finding in MDV algorithm is a two step process. In the first step, this algorithm calculates distance between objects, forming distance matrix and then it finds k-distance and k-distance neighborhood. The time complexity of this first step is well analyzed in [4]. If n is the cardinality and d is the dimensionality of database D and Let k be user supplied value (MinPts), then the time complexity of the first step is $O(n^2d + n k^2d)$ i.e. $O(n^2)$. Hear data is sequentially scanned.

In the second step k -distance variance is calculated and its time complexity is $O(n)$. So overall worst case time complexity is $O(n^2)$ which is comparable to LOF, LOF', LOF''.

MDV requires only two passes of the database D while, as mentioned in [1], LOF and LOF'' requires three passes and LOF' requires only two passes. Thus, it is clear that MDV needs less no. of passes than LOF and LOF''.

5 Experimental Results

In this section we present experiments that were done on some dataset obtained from the UCI Repository [7]. From this dataset, we selected 1000 records to create a test dataset for our experiment. Each record has 38 attributes. Out of these 1000 records in the test dataset there are 43 outliers. Experiments are conducted on a workstation on Pentium-4, 3 GHz processor and 1 GB of RAM. All algorithms are implemented in JAVA.

LOF is a well known outlier detection method and our algorithm is also based on LOF. So we compared our algorithm with LOF and with its variants (LOF' and LOF''). Proposed algorithm was executed for different input parameters. It was found that this algorithm is detecting outliers more effectively in most of the cases. Summary of the results after running all the algorithms is shown in Table 1. In this experiment, we considered MinPts1 = 10 and MinPts2 = 5 for LOF'' and MinPts = 10 for the rest.

Table 1. Results with MinPts1 = 10 and MinPts2 = 5 FOR LOF'' AND MinPts = 10 for the Rest

T	LOF	LOF'	LOF''	MDV
10	6	6	6	10
20	9	9	8	15
30	13	12	10	22
40	16	16	15	28
50	18	17	21	32
60	20	20	23	34
70	22	20	25	36
80	22	21	28	36
90	25	23	29	36
100	25	25	30	37

In table 1, T indicates the total number of outliers detected including correct and incorrect and entries for corresponding algorithm, indicates actual detection made by that algorithm. It is clear from the table that our algorithm MDV is detecting outliers efficiently in comparison to LOF, LOF' and LOF''. In fig.3, Curve is shown for the correct detection of the above four algorithm w.r.t. the total detected outliers.

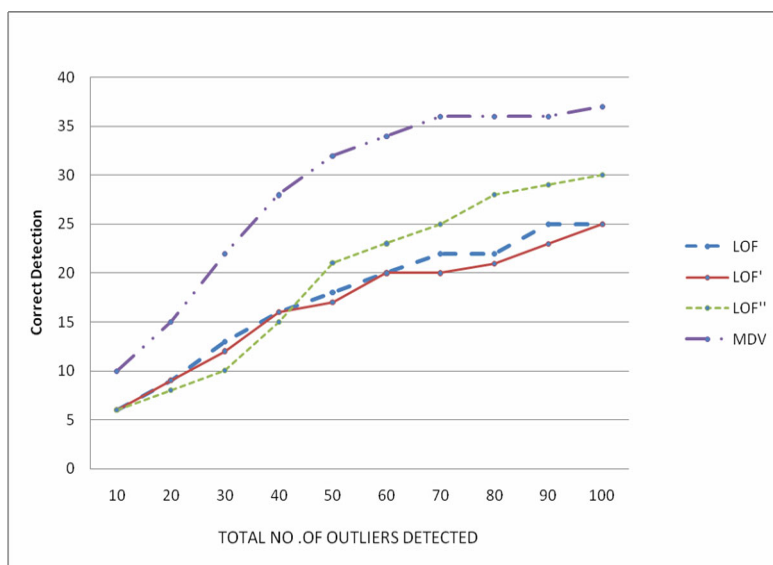


Fig. 3. Actual Outliers detected by MDV, LOF, LOF' and LOF''

6 Conclusion

In this paper we proposed a scheme for outlier detection that is based on LOF. We have shown that this new scheme is better than LOF and its variants (LOF' and LOF''). In contrast to calculate the LOF that is the ratio of densities, we considered the variance of densities and this is far better as is depicted in results. No. of passes required by our algorithm is less than that of LOF.

References

1. Chiu, A.L.-M., Fu, A.W.-C.: Enhancements on Local Outlier Detection. In: Proceedings of the Seventh International Database Engineering and Applications Symposium, IDEAS 2003 (2003)
2. Aggarwal, C.C., Yu, S.P.: An effective and efficient algorithm for high-dimensional outlier detection. The VLDB Journal 14, 211–221 (2005)
3. Hawkins, D.: Identification of Outliers. Chapman and Hall, London (1980)

4. Breunig, M.M., Kriegel, H.-P., Ng, R.T., Sander, J.: LOF:Identifying density-based local outliers. In: Proceedings of ACM SIGMOD International Conference on Management of Data, Dallas, Texas, USA, pp. 93–104 (2000)
5. Ester, M., Kriegel, H., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD 1996), Portland, Oregon, pp. 226–231 (1996)
6. Ankerst, M., Breunig, M.M., Kriegel, H.-P., Sander, J.: OPTICS: Ordering points to identify the clustering structure. In: Proceedings of ACM SIGMOD International Conference on Management of Data, Philadelphia, Pennsylvania, U.S.A, pp. 49–60 (1999)
7. Frank, A., Asuncion, A.: UCI Machine Learning Repository. University of California. School of Information and Computer Science, Irvine, CA (2010), <http://archive.ics.uci.edu/ml>
8. Han, J., Kamber, M.: Data Mining, Concepts and Techniques. Morgan Kaufmann, San Francisco (2001)
9. Angiulli, F., Pizzuti, C.: Outlier Mining in Large High-Dimensional Data Sets. IEEE Transactions on Knowledge and Data Engineering 17(2) (February 2005)
10. <http://en.wikipedia.org/wiki/Variance>

Hierarchical Clustering of Projected Data Streams Using Cluster Validity Index

Bharat Pardeshi and Durga Toshniwal

Department of Electronics and Computer Engineering
Indian Institute of Technology Roorkee
Roorkee - 247667, India
bhaarat001@gmail.com, durgafec@iitr.ernet.in

Abstract. Clustering is an unsupervised learning process of grouping a set of objects into classes of similar objects. Hierarchical method of clustering is an important data mining technique. In this paper we propose hierarchical clustering of projected data stream objects. Cluster Validity Index is used to accurately identify the desired number of clusters present in data set. Thus the user does not need to have prior knowledge about the number of classes present in given data stream. A multi-dimensional grid data structure is maintained, where the received data stream objects are projected. Using a fading function the data objects present in certain time limits are maintained, rest are discarded as time advances. Hierarchical clustering is then performed on this projected grid structure which gives the real clusters present in the given data stream at that instant of time. The proposed algorithm is fast enough to cope-up with the high speed stream as it just needs to find the connected cells present in the grid structure to discover clusters. Experiments performed on the synthetically generated data stream at the rate of 1000 records per second show that the results obtained reflect the actual cluster present.

Keywords: Hierarchical, Cluster Validity Index, Multi-dimensional Grid Structure, Projected Clustering.

1 Introduction

Clustering is an unsupervised learning process of grouping a set of objects into classes of similar objects [1]. Thus a cluster is collection of data objects that are similar to one another within the same cluster and dissimilar to objects in other clusters. Data Stream cluster analysis has many applications including surveillance systems, medical monitoring of life signs, communication networks, Internet traffic, and online transactions in the financial market or retail industry [2,3].

Data Streams are temporally ordered, fast-changing, massive, and potentially infinite volume of data [1]. This infinite data volume make it essential for the data stream clustering algorithms to process the arriving data objects in a single pass as it is difficult to store these high speed data stream objects. The developed

algorithm also needs to be fast enough to match the fast arriving rate of stream data. Because of the varying nature of data stream object attribute values, the clusters obtained should not be dominated by outdated stream data. For this we need to discard data stream objects while are older than particular time duration. Thus to discover knowledge or patterns from data streams, it is necessary to develop single-scan, on-line, multilevel, multidimensional stream processing and analysis algorithms.

The remainder of this paper is organized as follows. Section 2 contains the brief explanation about different data stream clustering algorithms proposed so far. Section 3.1 explains different clustering approaches with example. In Section 3.2 the cluster validity index is explained. In Section 4 we present the proposed clustering algorithm. We have given the obtained results in Section 5 and concluded the entire work in Section 6.

2 Related Work

Guha et al. [3] studied clustering data streams using K-median technique. This algorithm makes a single pass over the data stream and uses small space. The algorithm starts by clustering a calculated size sample according to the available memory into $2k$, and then at a second level, the algorithm clusters the above points for a number of samples into $2k$ and this process is repeated to a number of levels, and finally it clusters the $2k$ clusters into k clusters.

Babcock et al. [3] used exponential histogram (EH) data structure to improve Guha et al. algorithm. They use the same method described above, however they address the problem of merging clusters when the two sets of cluster centers to be merged are far apart by maintaining the EH data structure.

Charikar et al. [3] proposed another k-median algorithm that overcomes the problem of increasing approximation factors in the Guha algorithm with the increase in the number of levels used to result in the final solution of the divide and conquer algorithm.

Ordonez [3] proposed several improvements to k-means algorithm to cluster binary data streams. He developed an incremental k-means algorithm. He has demonstrated experimentally that the proposed algorithm outperforms the scalable k-means in the majority of cases. The use of binary data simplifies the manipulation of categorical data and eliminates the need for data normalization. The main idea behind the proposed algorithm is that it updates the cluster centers and weights after examining a batch of transactions which equalizes square root of the number of transactions rather than updating them one by one.

O'Challaghan et al. proposed STREAM [1,4,5] and LOCALSEARCH algorithms for high quality data stream clustering. The STREAM algorithm starts by determining the size of the sample and then applies the LOCALSEARCH algorithm if the sample size is larger than a pre-specified equation result. This process is repeated for each data chunk. Finally, the LOCALSEARCH algorithm is applied to the cluster centers generated in the previous iterations.

Aggarwal et al. proposed a framework for clustering data steams called CluStream[1,4,6] algorithm. This technique divides the clustering process into two

components. The online component stores summarized statistics about the data streams and the offline one performs clustering on the summarized data according to a number of user preferences such as the time frame and the number of clusters. They have recently proposed HPStream[4,7] a projected clustering for high dimensional data streams. HPStream has outperformed CluStream in recent results.

Gaber et al. [8,9] developed Lightweight Clustering LWC. It is an AOG-based algorithm. The algorithm adjusts a threshold that represents the minimum distance measure between data items in different clusters. This adjustment is done regularly according to a pre-specified time frame. It is done according to the available resources by monitoring the input-output rate. This process is followed by merging clusters when the memory is full.

3 Background Study

The dataset to be clustered contains n objects, which may represent persons, documents, countries and so on. In order to compute clusters of objects we require an appropriate clustering algorithm suitable for given data set. The cluster validity index is used to determine the number of classes present in the data set.

3.1 Clustering Approaches

There are many clustering methods available, and each of them may give a different grouping of a dataset. The choice of a particular method will depend on the type of output desired, including memory and time constraints. Typical clustering analysis methods are partition based clustering, hierarchical clustering, density based clustering, grid based clustering and model based clustering [1].

Partitioning methods construct k partitions (clusters) of the given dataset, where each partition represents a cluster. Each cluster may be represented by a centroid or a cluster representative which is some sort of summary description of all the objects contained in a cluster. K-means [10, 11] and K-medoids are the best examples of partitioning methods. Both the k-means and k-medoids algorithms are partitional (breaking the dataset up into groups) and both attempt to minimize squared error. In the k-means clustering problem, the centroid is not contained in the original points in most cases. In contrast to the k-means algorithm k-medoids chooses data points as centers which make k-medoids method more robust than k-means in the presence of noise and outliers, because a medoid is less influenced by outliers or other extreme values than a mean.

Hierarchical clustering [1] proceeds successively by either merging smaller clusters into larger ones, or by splitting larger clusters. A hierarchical method can be classified as being either agglomerative or divisive, based on how the decomposition is formed. The agglomerative approach starts with each object forming a separate group. It successively merges the objects or groups that are close to one another, until a desired number of clusters are obtained. The divisive approach, also called the top-down approach, starts with all of the objects in

the same cluster. In every successive iteration a cluster is split up into smaller clusters, until a desired number of clusters are obtained. However in hierarchical methods once a step (merge or split) is done, it can never be undone sometimes which may lead to erroneous decisions. DIANA [1] and AGNES [1] are typical examples of hierarchical clustering.

Density based clustering [1, 12] methods works on a local cluster criterion. Clusters are regarded as regions in the data space in which the objects are dense, and which are separated by regions of low object density (noise). Their general idea is to continue growing the given cluster as long as the density (number of objects or data points) in the neighborhood exceeds some threshold; that is, for each data point within a given cluster, the neighborhood of a given radius has to contain at least a minimum number of points. These regions thus formed may have an arbitrary shape. DBSCAN is a typical density based method that grows clusters according to a density-based connectivity analysis. OPTICS is another density-based method that generates an augmented ordering of the clustering structure of the data.

Grid-based clustering algorithm [1, 13] divides the multi-dimensional data space into a given number of cells, and then clustering operation is put on it. The main advantage of this approach is its fast processing time, which is typically independent of the number of data objects and dependent only on the number of cells in each dimension in the quantized space. STING (STatistical INformation Grid) [1, 14] is a typical example of a grid-based method based on statistical information stored in grid cells. It divides data space into rectangular cells, and these cells constitute a hierarchical structure: high-level cells can be divided into a number of low-level cells. The data statistical information (for example, mean, maximum, minimum, count and data distribution, etc.) of each cell is pre-calculated for the subsequent query processing.

WaveCluster and CLIQUE (CLustering In QUEst) [1, 14] are two clustering algorithms that are both grid-based and density-based. CLIQUE is an integrated algorithm based on density and grid. The idea is to divide M-dimensional data space into rectangular cells. If the number of data points in a cell is greater than a threshold (user input), it is called a dense cell. A cluster refers to the largest collection of dense cells. CLIQUE algorithm can automatically identify high dimensional space with high dense data points, and it is independent of data distribution and data input order.

In Model-based approach a model is hypothesized for each of the clusters and tries to find the best fit of that model to each other, e.g. EM, SOM, COBWEB. Another approach, Frequent pattern-based uses the analysis of frequent patterns, e.g. pCluster which performs clustering by pattern similarity. User-guided or constraint-based clustering approach considers user-specified or application-specific constraints, e.g. COD (obstacles), constrained clustering.

3.2 Cluster Validity Index

To obtain the natural clusters in the data set the user need to specify the number of desired clusters depending on the actual classes present in data set objects.

But it is difficult to determine the number of clusters k , if the user doesn't have any prior knowledge about the data set objects. A technique to solve this problem is to define a cluster validity index [9], which is used to estimate the optimal number of clusters. The Xie-Beni index (V_{xb}) [15] used to determine the cluster validity index which is the ratio of the average intra-cluster compactness to inter-cluster separation is

$$V_{XB} = \frac{\text{compactness}}{\text{separation}} = \frac{\frac{1}{n} \sum_{i=1}^k \sum_{j=1}^n \mu_{ij} \|x_j - v_i\|^2}{\min_{i \neq j \in \{1, \dots, k\}} \|v_j - v_i\|^2} \tag{1}$$

where $\|\cdot\|$ is the usual Euclidean norm, x_j is the j^{th} data object, v_i is the i^{th} clustering center and μ_{ij} is a Boolean value, which is equal to 1 if x_j is in cluster i , otherwise equal to 0. The best clustering result is the one in which the average intra-cluster variance is minimum and at the same time average inter cluster variance is maximum. Thus the value of k which gives minimum V_{xb} will decide upon the optimal k^* . The algorithm for finding optimal clustering number k^* is based on exhaustive search. Generally we iterate from $k_{min}=2$ to some clustering number k_{max} generally k_{max} is very less than the number of objects n in data set.

4 Proposed Work

In the given section we propose an overall framework for hierarchical clustering of projected data streams using cluster validity index. Due to the dynamic and fast changing nature of data streams, stream objects cannot be stored or accessed multiple times. Here we propose an algorithm for clustering which processes the stream objects in single pass with no need to store the actual data objects.

4.1 Maintenance of Projected Multi-dimensional Grid Structure

Assuming an n -dimensional data stream object, we will maintain an n -dimensional grid structure. Each dimension of this grid structure will represent an attribute of given stream object. The projection from each dimension will form a hyper-cube referred as Cell. As soon as a data stream object arrives we will project it in appropriate cell depending on attribute values. A cell will actually hold the count of number of objects that have attribute values matching to the given cell. Thus a cell can contain any number of data stream objects above the given minimum threshold value. A time-stamp is associated with each object while projecting into a cell, so as to fade out outdated history data as time passes. Also if any cell contains, less than given number of objects, then we will discard that cell as this cell will be holding outlier objects. Using this approach we don't actually need to store the data stream objects, instead we store the count of objects falling in that cell range.

Algorithm

1. Divide the each dimension of grid structure into n number of intervals. The projections from these divided intervals will form a virtual cell in n -dimensional space.
2. Whenever a data stream object s_i having n attributes arrives, project it into corresponding cell. Increment the count of objects present in that cell.
3. Fade out data stream objects which are outside our defined time limit.
4. Discard Cells which are having objects less than certain pre-defined threshold.

4.2 Proposed Hierarchical Clustering of Multi-dimensional Grid Structure

Whenever there is a request for clustering data stream, we will apply the hierarchical clustering on the previously maintained multi-dimensional grid structure. The beauty of this approach is that we don't need to calculate distances between any objects. The idea is to just go on connecting cells which are close to each other, until finally optimal numbers of clusters are discovered. Upon obtaining clusters from these connected cells, we will apply the cluster validity index to verify the validity of obtained clusters. The values of cluster validity index obtained for different k are stored in *validityIndexArray*. This step is repeated for given k_{max} to k_{min} numbers of clusters. The least value of validity index obtained from the discovered ranges of cluster stored in *validityIndexArray*, will give us the actual clusters present in the given data stream.

Algorithm

1. Start by assigning each cell to a cluster, so that if there are n cells in the projected grid structure, there are n clusters, each containing just one cell.
2. Find the cells which are adjacent to each other and merge them into a single cell of size which is equal to the sum of these cells.
3. Calculate the cluster validity index for the clusters obtained in *step 2* and store the value into *validityIndexArray*.
4. Repeat *steps 2 and 3* for k_{max} to k_{min} number of iterations.
5. The minimum value of validity index will give us the best clustering result.

Although we are evaluating the cluster validity index at each successive iteration, any additional computation is not required for different set of clusters obtained. While discovering clusters from k_{max} to k_{min} we will evaluate and store the cluster validity index for given number of k_i cluster. The minimum value of cluster validity index obtained for k_d (clusters desired) number of clusters will give the actual clusters present in data stream at that instant of time.

5 Experimental Results

The algorithm has been implemented in C language and all the experiments are performed on an Intel Core2 Duo T8100 processor having 3 GB of memory.

Table 1. Number of objects generated at different instance of time

	<i>Cluster1</i>	<i>Cluster2</i>	<i>Cluster3</i>
<i>Instance1</i>	330	446	204
<i>Instance2</i>	400	351	249
<i>Instance3</i>	375	419	206
<i>Instance4</i>	450	390	260

For experimental purpose we have taken the data stream objects having three dimensions. But this can be extended to n number of dimension. The data stream records were generated at the rate of 1000 records per second and a constant window size of 1000 records. The stream objects were generated randomly at different time instant in such a manner that it contained 3 inherent classes of data objects, each having given number of objects as shown in below table.

The clustering request was triggered immediately after the generation of mentioned objects at the specified 4 instances.

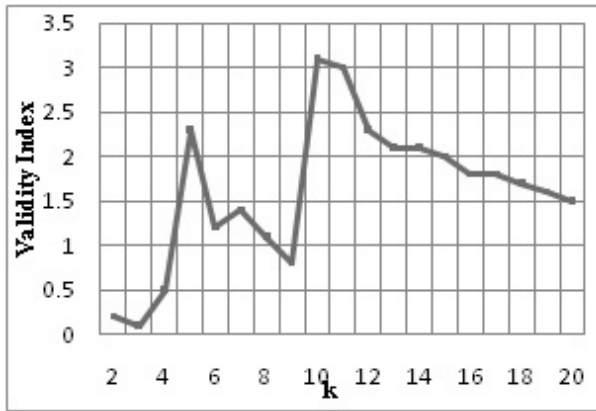


Fig. 1. Curve of Cluster Validity Index

The curve of cluster validity index obtained for $k=2$ to $k=20$ is as shown in Fig. 1. It can be observed that minimum value of cluster validity index is obtained at $k=3$, which implies that there are three classes of data objects present in stream.

It can be observed from Fig. 2, that there are three clusters discovered. At instance 1, Cluster 1 contains 330 objects, cluster 2 contains 466 objects and cluster 3 contains 204 objects. At instance 2, cluster 1 contains 400 objects, cluster 2 contains 351 objects and cluster 3 contains 249 objects. The change of

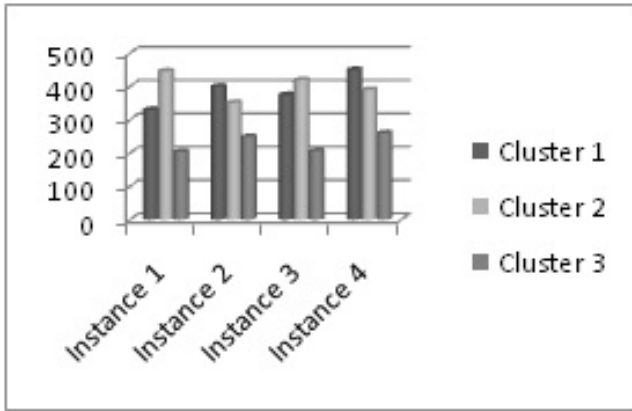


Fig. 2. Obtained Cluster Information

trend can be observed in the successive plotted cluster information at instance 3 and instance 4.

6 Conclusion

A hierarchical clustering of projected data streams using cluster validity index has been proposed. By making use of cluster validity index, the user does not require to have the prior knowledge about number of clusters present in the data stream. This is very essential especially for data stream clustering where the nature of data stream keeps on varying with respect to time. In the proposed technique there is no need to actually store data stream objects, they are accessed only once and then projected into appropriate cell. Each cell holds the count of objects falling in given attribute range of that cell. Also we have applied hierarchical clustering, so we don't need to evaluate the obtained cluster from scratch for calculating cluster validity index. This was only possible in hierarchical clustering because in hierarchical clustering we perform bottom up clustering from k_{max} to k_{min} set of clusters. Experiments performed on synthetic data set show that this algorithm can accurately find meaningful clusters in the given data stream with any arbitrary shape. Thus our proposed work satisfies all the essential requirements of data stream clustering i.e. single-pass, fast processing, dynamic and minimum memory requirements.

References

1. Han, J., Kamber, M.: Data Mining: Concepts and Techniques, 2nd edn. Elsevier Inc., Rajkamal Electric Press (2006)
2. Thakkar, H., Mozafari, B., Zaionolo, C.: A Data Stream Mining System. In: IEEE Conference on Data Mining Workshops. University of California, Los Angeles (2008)

3. Ikonovska, E., Loskovska, S., Gjorgjevik, D.: A Survey of Stream Data Mining. In: Eighth National Conference with International Participation - E/TAI, Ohrid, Republic of Macedonia (2007)
4. Aggarwal, C.C.: Data Streams Models and Algorithms, 1st edn. Springer Publications, Heidelberg (2007)
5. Guha, S., Meyerson, A., Mishra, N., Motwani, R., O'Callaghan, L.: Clustering Data Streams: Theory and Practice. TKDE special issue on clustering 15 (2003)
6. Aggarwal, C., Han, J., Wang, J., Yu, P.S.: A Framework for Clustering Evolving Data Streams. In: International Conference on Very Large Databases, Berlin, Germany (2003)
7. Aggarwal, C., Han, J., Wang, J., Yu, P.S.: A Framework for Projected Clustering of High Dimensional Data Streams. In: Proceedings of 30th VLDB Conference, Toronto, Canada (2004)
8. Gaber, M.M., Krishnaswamy, S., Zaslavsky, A.: Cost-Efficient Mining Techniques for Data Streams. In: Conferences in Research and Practice in Information Technology, Dunedin, New Zealand (2004)
9. Gaber, M.M., Krishnaswamy, S., Zaslavsky, A.: Resource-aware Mining of Data Streams. Journal of Universal Computer Science 11(8) (2005)
10. Gu, J., Chen, X., Zhou, J.: An Enhancement of K-means Clustering Algorithm. In: The Second International Conference on Business Intelligence and Financial Engineering, Beijing, China (2009)
11. Kanungo, T., Mount, D.M., Netanyahu, N.S., Piatko, C.D., Silverman, R., Wu, A.Y.: An Efficient k-Means Clustering Algorithm: Analysis and Implementation. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(7) (2002)
12. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In: Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining, München, Germany (1996)
13. Liao, W., Liu, Y., Choudhary, A.: A Grid-based Clustering Algorithm using Adaptive Mesh Refinement. In: 7th Workshop on Mining Scientific and Engineering Datasets, Florida (2004)
14. Zăiane, Ó.R.: Introduction to Data Mining. Technical Report, Principles of Knowledge Discovery in Databases, CMPUT690, University of Alberta (2009)
15. Xie, J., Zhang, Y., Jiang, W.: A K-means Clustering Algorithm with Meliorated Initial Centers and Its Application to Partition of Diet Structures. In: International Symposium on Intelligent Information Technology Application Workshops, pp. 98–102 (2008)

Non-Replicated Dynamic Fragment Allocation in Distributed Database Systems

Nilarun Mukherjee

Senior Lecturer, Department of CSE/IT
Bengal Institute of Technology, Kolkata, West Bengal, India
nilarun.mukherjee@gmail.com

Abstract. Distributed databases have become the most essential technology for recent business organizations. The performance, efficiency and availability of a Distributed Database largely depend on the fragmentation of global relations and the allocation of those fragments at several sites of the network. In Non-Replicated fragment allocation scenario, the optimum allocation of the fragments is the only way, which can be exploited to increase the performance, efficiency, reliability and availability of the Distributed Database. In this paper, a new dynamic fragment allocation algorithm is proposed in Non-Replicated allocation scenario, which incorporates access threshold, time constraints of database accesses and most importantly the volume of data transmitted to dynamically reallocate fragments to sites at runtime in accordance with the changing access probabilities of nodes to fragments. It will decrease the migration of fragments and data transfer cost and will also improve the overall performance by dynamically reallocating fragments in a most intuitive manner.

Keywords: Distributed Database, Fragmentation, Dynamic Fragment Allocation, Local Agent, Root Agent.

1 Introduction

A Distributed Database is a collection of data, which logically belong to the same system, but are distributed over the sites of a computer network. Each site of the network has autonomous processing capability and can perform local database applications [9]. Each site also participates in the execution of at least one global database application, which requires accessing data residing at several different sites from geographically dispersed locations, using a communication subsystem.

The primal motivations for distributed databases are to improve performance, to increase the availability of data, shareability, expandability and access facility. In a distributed database, maximization of the locality of processing of database applications by allocating data as close as possible to the applications which use them significantly reduces the communication overhead with respect to a centralized database.

In most of the cases Distributed Databases are developed in a Bottom-Up approach, where there already exist several centralized databases located at several geographically dispersed sites, which are integrated to form the Distributed Database. In this scenario, there is no scope of fresh fragmentation, as the fragments exist beforehand and they are

analyzed and integrated to form the global relations. Moreover, fragmentation depends highly on the business and organizational requirements. Therefore, the optimum allocation of those fragments is the only way, which can be exploited by the designers to increase the performance, efficiency, reliability and availability of the Distributed Database. The main aim is to maximize the locality of processing for each distributed application by storing the fragments closer to where they are more frequently used in order to achieve best performance. This means placing data i.e. fragments required by the distributed applications at their site of origin or at sites which are closer to their site of origin. I.e. to maximize the “local” references and to minimize “remote” references to the data by each distributed application with respect to its site of origin [9]. This is done by adding the number of local and remote references corresponding to each candidate fragmentation and fragment allocation, and selecting the best solution among them, i.e. the solution providing highest processing locality or Complete Locality to maximum number of distributed applications [9]. The *Complete Locality* means the distributed application can be completely executed at its site of origin; thus reducing overall remote accesses.

Various approaches have already been evolved for dynamic allocation of data in distributed database [1] to improve the database performance. These address the more realistic dynamic environment, where the access probabilities of nodes to fragments change over time. Fragment allocation can further be divided in two ways: Non – Redundant Fragment Allocation: Each fragment of each global relation is allocated to exactly one site. Redundant Fragment Allocation: Fragments of each global relation are allocated to one or more sites introducing replication of the fragments.

In this paper, a new dynamic fragment allocation algorithm is proposed in Non-Replicated allocation scenario, which incorporates access threshold, time constraints of database accesses and most importantly the volume of data transmitted to dynamically reallocate fragments to sites at runtime in accordance with the changing access pattern i.e. the changing access probabilities of nodes to fragments over time. The proposed algorithm will decrease the movement of fragments over the network and data transfer cost and will also improve the overall performance of the system by dynamically allocating fragments in a most intuitive manner.

2 Related Work

Till date many works have been published on the problem of allocation of data or fragments of the global relations to geographically dispersed sites of a distributed database. [10] considered the problem of file allocation for typical distributed database applications with a simple model of transaction execution. [11] incorporated issues like concurrency and queuing costs. [12] provides an integrated approach for fragmentation and allocation. [12] identified seven criteria that a system designer can use to determine the fragmentation, replication and allocation. [15] presents a replication algorithm that adaptively adjusts to changes in read-write patterns. [16] provides an approach based on Lagrangian relaxation and [17] describes heuristic approaches. More recently [18] has given a high-performance computing method for data allocation in distributed database system.

In most of the above approaches, data allocation has been proposed prior to the design of the distributed database, depending on some static data access patterns and/or query patterns. Static allocation of fragments provides the best solution when the access probabilities of nodes to fragments never change over time, but, degrades performance in a dynamic environment, where probabilities change over time.

Over past few years, work has been introduced for dynamic fragment allocation in distributed database systems. [15] gives a model for dynamic data allocation for data redistribution and incorporates a concurrency mechanism. In [4] an algorithm is proposed for dynamic data allocation, which reallocates data with respect to the changing data access patterns. [13] provides approach for allocating fragments by adapting a machine learning approach. [7] considers incremental allocation and reallocation based on changes in workload. [19] incorporated security considerations into the dynamic file allocation process. In [20] an optimal fragment allocation algorithm for non-replicated distributed database systems is proposed. [21] has introduced a threshold algorithm for non-replicated fragment allocation in distributed databases. In the threshold algorithm, the fragments are continuously reallocated according to the changing data access patterns.

In this paper, a new dynamic fragment allocation algorithm is proposed in Non-Replicated allocation scenario which is an extension of the work carried out by [1] and [4, 21, 6]. The proposed algorithm incorporates access threshold, time constraints of database accesses and most importantly the volume of data transmitted to dynamically reallocate fragments to sites at runtime in accordance with the changing access pattern i.e. in accordance with the changing access probabilities of nodes to fragments over time.

3 Design

The main factor that affects the efficiency and turnaround time of the applications of a distributed database is the time delay for transferring the data needed by a certain distributed database query or application over the network from the fragments located at several different geographically dispersed / remote sites to the site of origin of the application or query. Therefore, the primary aim is to allocate fragments accessed or needed by certain distributed database application either at the same site where the distributed database application / query is invoked or at sites closer to the site of origin, so that the data transmission over the network is minimized during the execution of the distributed database application / query. On the other hand, it is not a feasible solution to place the entire data of a distributed database at every site of the system; also it will violate the basic requirements of a distributed database [9]. The placing of fragments to the same sites or to the sites which are closer to the sites, where from the distributed database applications needing those fragments are invoked, is a considerably complex problem. Especially, in the scenario where the probability or pattern of accessing fragments by the distributed database applications at different sites changes dynamically over time.

The proposed algorithm for dynamic fragment allocation in distributed database exploits the concepts of the existing algorithms: the Optimal Algorithm [4], the Threshold Algorithm [21] and TTCA Algorithm [1]. In distributed database the efficiency and

turnaround time of the distributed database applications ultimately depends on the volume of data that is required to get transferred over the network from one site to another for the execution of those applications. Reallocating a fragment from a certain site to another site, which makes the highest number of accesses to that fragment and not to the site, which even though does not make the highest number of accesses to that fragment but results in transmission of maximum volume of data from or to that fragment, will not be always beneficial i.e. will not always improve the overall performance or efficiency of the distributed database.

The main concept of this work is to reallocate a fragment located at a certain site to another site, which not only makes highest number of accesses to that fragment in a specific period of time but also results in transmission of maximum volume of data from or to that fragment in that specific period of time. Moreover, most of the time the site which makes the highest number of accesses to a particular fragment located at a particular site in a specific period of time also results in transmission of maximum volume of data from or to that fragment in that specific period of time. Thus, for most of the cases the proposed algorithm for dynamic fragment allocation will reflect the most realistic scenario and will result in more intuitive and optimum dynamic fragment reallocation.

Furthermore, reallocation of fragments containing huge amount of data from one site to another site over the network just in order to make the distributed database applications to have their required data at their site of origin or at the sites very closer to their site of origin, incurs huge data transmission over the network and results in transmission overload. Because, fragments are expected to contain huge amount of data as compared to the data required by and transmitted due to the execution of those distributed database applications. Thus, it is not desirable to have frequent migrations of fragments over the network, as this can significantly affect or degrade the overall performance or efficiency of the distributed database. The proposed algorithm improves the overall performance or efficiency of the distributed database by imposing a more strict condition for fragment reallocation and results in fewer migrations of fragments from one site to other over the network during the execution of the distributed database application / query, in Non-Replicated allocation scenario as compared to the Optimal Algorithm [4], the Threshold Algorithm [21] and TTCA Algorithm [1].

Algorithm:

1. Initially all the fragments of all the global relations of the distributed database are distributed over different sites using any static allocation method in non-replicated manner. Let there be total N number of fragments of global relations distributed among the total M number of sites in the distributed database system. Each site has one or more fragments allocated to it. A distributed database query or application may require accessing several different fragments allocated at several different sites for execution.
2. The proposed algorithm needs each site to maintain a separate data structure named Access Log, which stores certain information regarding each access to the fragments allocated at that site, by the distributed database queries or applications invoked at the same or different sites. Each Access Log record denoted by A_k^h (i.e. k^{th} access at site h , where $k = 1,2,3,\dots$ to infinity and $h = 1,2,3,\dots,M$) stores the following information regarding each access:

- a. Name or Identifier of the Fragment accessed.
 - b. Address of the accessing site.
 - c. Date and Time of the access.
 - d. The volume of data transmitted from or to that fragment in Bytes.
3. Two parameters are used to tune the performance of the algorithm:
- a. The Time Constraint for Fragment Reallocation (τ). It determines the duration of the time intervals.
 - b. The Access Threshold for Fragment Reallocation (η).

The following Steps [4 to 6] are performed at each site for each individual access to a fragment allocated at that site by a certain distributed database query or application invoked at the same or different site. Suppose at site h an access is made and processed for fragment i allocated at that site from site j at time t , where $h = 1, 2, 3, \dots, M$, $i = 1, 2, 3, \dots, N$ and $j = 1, 2, 3, \dots, M$ and $h = j$ or $h \neq j$. The local agent at site h performs the following operations:

4. Write a log record A_k^h in Access Log at site h .
5. If the address of the accessing site in the log record A_k^h is same as the address of site h , i.e. the access is made from the same site ($h = j$), then do nothing.
6. Else if the address of the accessing site in the log record A_k^h is different from the address of site h , i.e. the access is made from a different site ($h \neq j$); then:
 - a. Calculate the total number of accesses made from all the sites (including site h) to the fragment i located at site h within the time interval τ up to current access time t . Let n_i^m , denotes the total number of accesses from the site m to the fragment i allocated at site h , within the time interval τ up to current access time t , where $m = 1, 2, 3, \dots, M$.
 - b. Calculate the Average volume of data transmitted (in bytes) in between the fragment i and all the sites (including site h) where from the accesses to the fragment i located at site h are made, through the accesses occurred within the time interval τ up to current access time t . Let $A_k^h V_i^m$ denotes the volume of data transmitted (in bytes) in between the fragment i allocated at site h and the site m in the access A_k^h at certain point of time, where $m = 1, 2, 3, \dots, M$. The Average volume of data transmitted in bytes (denoted by $V_i^{m,t}$) in between the fragment i allocated at site h and the site m through the accesses occurred within the time interval τ up to current access time t can be calculated as:

$$V_i^{m,t} = \frac{\sum A_k^h V_i^m}{n_i^j} \quad (1)$$

Where A_k^h occurred within the time interval τ up to current access time t . $V_i^{m,t}$ is calculated for all the sites (including site h) where from the accesses to the fragment h are made, through the accesses occurred within the time interval τ up to current access time t .

- c. If the total number of accesses from the site j to the fragment i allocated at site h , within the time interval τ up to current access time t is greater than the access threshold for Fragment Reallocation (η) and is greater than the total number of accesses made from all other sites (including site h) to the fragment i

located at site h within the time interval τ up to current access time t i.e. if $n_i^j > \eta$ and $n_i^j > n_i^m$ where $j, m = 1, 2, 3, \dots, M$ and $m \neq j$ and $h \neq j$, then the fragment i is migrated and reallocated to site j and removed from the current site h , catalogs are updated accordingly, if and only if the Average volume of data transmitted (in bytes) in between the fragment i allocated at site h and the site j through the accesses occurred within the time interval τ up to current access time t is greater than the Average volume of data transmitted (in bytes) in between the fragment i and all other sites (including site h) where from the accesses to the fragment i located at site h are made, through the accesses occurred within the time interval τ up to current access time t , i.e. if and only if $V_i^j t > V_i^m t$ where $j, m = 1, 2, 3, \dots, M$ and $m \neq j$ and $h \neq j$. Otherwise, do nothing.

This dynamic fragment allocation algorithm migrates a fragment located at a certain site to another site, which not only makes number of accesses to that Fragment greater than the Access Threshold for Reallocation (η) in the specific period of time determined by the Time Constraint for Fragment Reallocation (τ) up to current access time, but also results in transmission of maximum volume of data from or to that fragment in that specific period of time.

In distributed database, at each site the local agent of a remote or local distributed transaction accesses or interacts with the local database where the fragments of the global relations are actually stored [9]. In case of insertion or modification operations on a fragment, the local agent receives the data to be inserted or to be used to modify the existing data in the particular fragment from the root agent of the distributed transaction. Thereafter, the local agent sends those data along with the command (insert or update) to the local transaction manager, which actually accesses the database table corresponding to that fragment stored at the local database to perform insertion or modification. In case of retrieval, the local agent receives the data retrieved from the database table stored at the local database corresponding to a particular fragment through the local transaction manager. Thereafter, it sends those retrieved data to the root agent of the distributed transaction executing at the same or different site in the distributed database network. In all the cases the local agent of the distributed database transaction temporarily retains the data to be inserted or to be used to modify the existing data in the particular fragment or being retrieved from the particular fragment during the execution of a distributed database application or query. Moreover, the local agent belongs to the distributed database transaction management system and it is designed and programmed by the developer of the distributed database management system. Therefore, it is possible to enhance the transaction management system and the local agents of the distributed database transactions to add the functionality of writing Access Log Records and to calculate the volume of data transmitted from or to each fragment (in Bytes) at each fragment access, i.e. to implement the proposed dynamic fragment allocation algorithm.

4 Comparative Study

In Optimal Algorithm [4], initially all the fragments are distributed over the different sites using any static data allocation method. After the initial allocation, system maintains an access counter matrix for each locally stored fragment at each site or node.

Every time an access request is made for the locally stored fragment, the access counter of the accessing site for that fragment is increased by one. If the counter of a remote node becomes greater than the counter of the current owner, then the fragment is moved to the accessing node. The problem of Optimal Algorithm [4] technique is that if the changing frequency of access pattern for each fragment is high, then it will spend more time for transferring fragments to different sites and will incur huge data transmission overload. Threshold algorithm [21] solves the problem of optimal algorithm and decreases the migration of fragments over different sites and reduces the transmission overload as compared to the Optimal Algorithm [4]. Threshold algorithm guarantees the stay of the fragment for at least $(\eta + 1)$ accesses at the new node after a migration, where η is the value of threshold. The most important point in this algorithm is the choice of threshold value. If the threshold value increases then the migration of fragment will be less. But, if the threshold value decreases then there will be more migration of fragments. But threshold algorithm resets the counter of local fragment to zero, every time a node is going for a local access and it does not specify which node will be the fragment's new owner when the counter exceeds the threshold value, also it does not give the information about past accesses of the fragments and does not consider the time variable of the access pattern.

The TTCA [1] algorithm removes all the above problems of threshold algorithm [21] by adding a time constraint to consider the time of the accesses made to a particular fragment. TTCA [1] decreases the migration of fragments over different sites, thus reduces the transmission overload due to fragment migrations as compared to simple Threshold algorithm [21]. TTCA [1] maintains a counter matrix at each site to store the number of accesses made by the sites to a particular fragment located at that site, which is incremented on each access. It reallocates data with respect to the changing data access patterns with time constraint. It migrates a fragment from one site to another site, which has most recently accessed that fragment for $\eta + 1$ numbers of times in a time period τ up to the current access time, where η is the Threshold Value and τ is the Time Constraint. If the value of time constraint increases then the migration of fragment will be more. But, if the value of time variable decreases then there will be less migration of fragments. But TTCA [1] does not store the time of those accesses in that counter matrix. Therefore, it does not provide any conceivable method to determine within which time period the certain number of accesses (determined by the value of the counter) made by a certain site to a particular fragment located at another site have occurred.

The proposed algorithm will remove all the above problems of the Threshold [21] and TTCA [1] algorithm. It will dynamically reallocate fragments at runtime in accordance with the changing access pattern i.e. the changing access probabilities of sites to fragments over time. While migrating fragments from one node to another node, this algorithm not only considers the threshold value and time constraint of database accesses and the time of the accesses made to a particular fragment, but also considers the volume of data transmitted. This dynamic fragment allocation algorithm migrates a fragment located at a certain site to another site, which not only makes number of accesses to that fragment greater than the Access Threshold for Reallocation (η) in the specific period of time determined by the Time Constraint for Fragment Reallocation (τ) up to current access time, but also results in transmission of maximum volume of data from or to that fragment in that specific period of time. It recognizes the fact that reallocating a fragment

from a certain site to another site, which makes the highest number of accesses to that fragment and not to the site, which even though does not make the highest number of accesses to that fragment but results in transmission of maximum volume of data from or to that fragment, will not be always beneficial i.e. will not always improve the overall performance or efficiency of the distributed database. The proposed algorithm for dynamic fragment allocation reflects the most realistic scenario and results in the optimum dynamic fragment reallocation, in Non-Replicated allocation scenario. The proposed algorithm improves the overall performance or efficiency of the distributed database by imposing a more strict condition for fragment reallocation and resulting in fewer migrations of fragments from one site to other over the network and further reduces the transmission overload due to fragment migrations as compared to the Optimal Algorithm [4], the Threshold Algorithm [21] and TTCA Algorithm [1].

Moreover, the administrators of the distributed database implementing this dynamic fragment allocation algorithm can easily relax or tighten the condition for fragment reallocation by regulating the values of the Time Constraint for Fragment Reallocation (τ) and the Access Threshold for Fragment Reallocation (η), to tune the algorithm to work best to fulfill the functional, infrastructural and business requirements of the organization implementing the distributed database. The choice of the most appropriate values of τ and η serves as the key factors for determining the optimum performance of the algorithm, to minimize the frequency of unnecessary fragment migrations and to reduce the movement of data over the network required during the execution of the distributed database queries or applications by increasing their locality of processing i.e. to maximize the overall throughput, performance and efficiency of the distributed database applications. For a fixed value of τ , if the value of η is increased, then the migration of fragments from one site to other will decrease vice versa. For a fixed value of η , if the value of τ is increased, then the migration of fragments from one site to other will increase vice versa. I.e.

$$\frac{\text{Fragments Migration /}}{\text{Reallocation Frequency}} \propto \frac{\tau}{\eta} \quad (2)$$

The proposed algorithm requires more storage space and increases the storage cost due to the maintenance of the Access Log at each site, in which log records are created for each access. Although, it reduces some storage space requirement by eliminating the usage of the counter matrix at each site, but it incurs more computational overhead per database access at each site due to the calculation of the total number of accesses made from all other sites to a particular fragment located at that site within the time interval τ up to current access time and due to the calculation of the Average volume of data transmitted (in bytes) in between that fragment and all other sites where from the accesses to that fragment are made, through the accesses occurred within the time interval τ up to current access time. But these drawbacks are well compensated by the advantage gained from the reduction in the frequency of unnecessary fragment migrations as compared to the Optimal Algorithm [4], the Threshold Algorithm [21] and TTCA Algorithm [1].

5 Conclusion

In this paper a new sophisticated dynamic fragment allocation algorithm is proposed in Non-Replicated allocation scenario which is an extension of the work carried out by [1] and [4, 21, 6]. The proposed algorithm incorporates access threshold, time constraints of database accesses and most importantly the volume of data transmitted to dynamically reallocate fragments to sites at runtime in accordance with the changing access pattern i.e. in accordance with the changing access probabilities of nodes to fragments over time. This dynamic fragment allocation algorithm migrates a fragment located at a certain site to another site, which not only makes number of accesses to that Fragment greater than the Access Threshold for Reallocation (η) in the specific period of time determined by the Time Constraint for Fragment Reallocation (τ) up to current access time, but also results in transmission of maximum volume of data from or to that fragment in that specific period of time. It reflects the most realistic scenario and results in the more intuitive and optimum dynamic fragment reallocation, in Non-Replicated allocation scenario. The proposed algorithm improves the overall performance or efficiency of the distributed database by imposing a more strict condition for fragment reallocation in distributed database and resulting in fewer migrations of fragments from one site to other over the network as compared to the Optimal Algorithm [4], the Threshold Algorithm [21] and TTCA Algorithm [1]. Moreover, the administrators of the distributed database implementing this dynamic fragment allocation algorithm can easily relax or tighten the condition for fragment reallocation by regulating the values of the Time Constraint for Fragment Reallocation (τ) and the Access Threshold for Fragment Reallocation (η), to tune the algorithm to work best for the functional, infrastructural and business needs of the organization. The choice of the most appropriate values of τ and η serves as the key factors for determining the optimum performance of the algorithm, to minimize the frequency of unnecessary fragment migrations and to minimize the movement of data over the network required during the execution of the distributed database queries or applications by increasing their locality of processing i.e. to maximize the overall throughput, performance and efficiency of the distributed database applications.

References

1. Singh, A., Kahlon, K.S.: Non-replicated Dynamic Data Allocation in Distributed Database Systems. *IJCSNS International Journal of Computer Science and Network Security* 9(9), 176–180 (2009)
2. Vasileva, S., Milev, P., Stoyanov, B.: Some Models of a Distributed Database Management System with data Replication. In: *International Conference on Computer Systems and Technologies-CompSysTech 2007*, vol. II, pp. II.12.1—II.12.6 (2007)
3. Agrawal, S., Narasayya, V., Yang, B.: Integrating Vertical and Horizontal Partitioning into Automated Physical Database Design. In: *Proc. 2004, ACM SIGMOD International Conf. Management of Data*, pp. 359–370 (2004)
4. Brunstroml, A., Leutenegger, S.T., Simhal, R.: Experimental Evaluation of Dynamic Data Allocation Strategies in a Distributed Database with changing Workload. *ACM Trans. Database Systems* (1995)

5. March, S., Rho, S.: Allocating Data and Operations to Nodes in Distributed Database Design. *IEEE Trans. Knowledge and Data Eng.* 7(2), 305–317 (1995)
6. Ulus, T., Uysal, M.: A Threshold Based Dynamic Data Allocation Algorithm- A Markove Chain Model Approach. *Journal of Applied Science* 7(2), 165–174 (2007)
7. Chin, A.: Incremental Data Allocation and Reallocation in Distributed Database Systems. *Journal of Database Management* 12(1), 35–45 (2001)
8. Hanamura, H., Kaji, I., Mori, K.: Autonomous Consistency Technique in Distributed Database with Heterogeneous Requirements. In: *IPDPS Workshops 2000*, pp. 706–712 (2000)
9. Ceri, S., Pelegatti, G.: *Distributed Database Principles & Systems*. McGraw-Hill International Editions
10. Ceri, S., Martella, G., Pelagatti, G.: Optimal file Allocation for a Distributed Database on a Network of Minicomputers. In: *Proc. International Conference on Database*, Aberdeen, British Computer Society, Hayden (July 1980)
11. Ram, S., Narasimhan, S.: Database Allocation in a Distributed Environment: Incorporating a Concurrency Control Mechanism and Queuing Costs. *Management Science* 40(8), 969–983 (1994)
12. Karlaplem, K., Pun, N.: Query-Driven Data Allocation Algorithms for Distributed Database Systems. In: Tjoa, A.M. (ed.) *DEXA 1997*. LNCS, vol. 1308, pp. 347–356. Springer, Heidelberg (1997)
13. Tamhankar, A., Ram, S.: Database Fragmentation and Allocation: An Integrated Methodology and Case Study. *IEEE Trans. Systems, Man and Cybernetics Part A* 28(3) (May 1998)
14. Chaturvedi, A., Choubey, A., Roan, J.: Scheduling the Allocation of Data Fragments in a Distributed Database Environment: A Machine Learning Approach. *IEEE Trans. Eng. Management* 41(2), 194–207 (1994)
15. Wolfson, O., Jajodia, S., Huang, Y.: An Adaptive Data Replication Algorithm. *ACM Trans. Database Systems* 22(2), 255–314 (1997)
16. Chiu, G., Raghavendra, C.: A Model for Optimal Database Allocation in Distributed Computing Systems. In: *Proc. IEEE INFOCOM 1990*, vol. 3, pp. 827–833 (June 1990)
17. Huang, Y., Chen, J.: Fragment Allocation in Distributed Database Design. *J. Information Science and Eng.* 17, 491–506 (2001)
18. Hababeh, I.O., Ramachandran, M., Bowring, N.: A high-performance computing method for data allocation in distributed database systems. *J. Supercomput.* 39, 3–18 (2007)
19. Mei, A., Mancini, L., Jajodia, S.: Secure Dynamic Fragment and Replica Allocation in Large-Scale Distributed File Systems. *IEEE Trans. Parallel and Distributed Systems* 14(9), 885–896 (2003)
20. John, L.S.: *A Generic Algorithm for Fragment Allocation in Distributed Database System*. ACM, New York (1994)
21. 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)

A Comparative Study of Machine Learning Algorithms as Expert Systems in Medical Diagnosis (Asthma)

B.D.C.N. Prasad¹, P.E.S.N. Krishna Prasad², and Yeruva Sagar³

¹ Professor, Dept. of Computer Applications,
P.V.P. Siddartha Institute of Technology, Vijayawada- 520007
bdcnprasad@gmail.com

² Associate Professor, Dept. of CSE, Aditya Engineering College,
Kakinada, 533 437, India
surya125@gmail.com

³ Associate Professor, Dept. of CSE, St. Peters Engineering College, HYD, India
sagaryeruva@yahoo.com

Abstract. In Medical Diagnosis a plenty of complex diseases could not be predicted properly. Now a days in India one of the major diseases is Asthma. In order to diagnose the disease asthma with Expert Systems, identifying using Machine learning algorithms such as Auto-associative memory neural networks, Bayesian networks, ID3 and C4.5. We present a comparative study among these algorithms with the use medical expert systems on patient data. We gathered the clinical signs and symptoms of asthma of patients from various resources. Based on the analysis of such data, it is found that the Auto-associative memory neural networks are one of the best among the remaining algorithms in terms of efficiency and accuracy of identifying the proper disease.

Keywords: Expert Systems, Context-sensitive Auto-associative memory, C4.5, Neural Networks, Medical diagnosis, Asthma.

1 Introduction

The importance of asthma as a cause of chronic respiratory disease has increased. Asthma (AZ-ma) is a chronic (long-term) lung disease that inflames and narrows the airways. Asthma causes recurring periods of wheezing (a whistling sound when you breathe), chest tightness, shortness of breath, and coughing. The coughing often occurs at night or early in the morning. Asthma affects people of all ages, but it most often starts in child. This has initiated a large number of studies designed to detect risk factors for asthma. We have chosen two methodologies to diagnose the disease Asthma. First method, with the use of questionnaires, and the second one is clinical diagnosis. We developed expert systems with the help of Machine learning techniques such as Auto-associative memory neural networks, Bayesian networks, ID3 and C4.5 algorithms using Questionnaire approach. Currently we are planning to implement the second

approach with clinical data such as X-Rays, blood reports, to diagnose the stage of asthma using same algorithms.

A comparative study has been made on these algorithms using Questionnaire method. We have interacted with the expert Physicians in specialization repository system. They have given a list of questionnaire for identifying the disease asthma and asthma like symptoms from the information provided by the patients. We developed an Expert System from the given questionnaire and analysis can be made by applying machine algorithms to diagnose the Asthma and also find the appropriate algorithm that suits the effective outcome. From our study, we determined Auto-associative memory neural networks are one the best techniques to diagnose the asthma and asthma like Symptoms.

Validity and reliability are general problems with questionnaires. The reliability can readily be tested by administering the same questionnaire two or more times to the same individuals. The validation procedure is much more troublesome, as there is no generally accepted operational definition of asthma, on which operational definition of asthma has been used.

Causes of Asthma: The exact cause of Asthma isn't known. Researchers think a combination of factors (family genes and certain environmental exposures) interact to cause asthma to develop, most often early in life. Different factors may be more likely to cause asthma in some people than others. Researchers continue to explore what causes asthma.

Symptoms of Asthma: Symptoms of asthma may occur for no obvious reason. They may include: a) Feeling breathless (you may gasp for breath), b) a tight chest (like a band tightening around your chest), c) wheezing (a whistling sound when you breathe), and d) coughing, particularly at night.

The severity and duration of the symptoms of asthma are often variable and unpredictable, and are sometimes worse during the night or with exercise. The symptoms of a severe asthma attack often develop slowly, taking between 6-48 hours to become serious.

The Questionnaire will be like yes or No options for user convenience and also they can give clear information about symptoms which they are being suffered. This is a very much like Physicians preliminary diagnosis before going to clinical diagnosis. This expert system will be designed and implements for adults only and this may not be the exact preliminary diagnosis, but on different trails of individuals, we would like to tell this is one way to diagnose the Asthma. The list of questions will be presented below in Table 1, given Expert Physicians.

Table 1. Questions About Asthma and Asthma-like Symptoms in Respiratory System

1. Is breathing problem Episodic or Paroxysmal Intermittent?
2. Is breathing problem associated with Cough?
3. Do you feel tightness of Chest?
4. Is breathing problem associated with Wheeze? Does the problem

occurs seasonal?

5.Does the problem occur early in the mornings or nights?

6.Does the problem occurs or becomes sever in Cold Air?

7.Does the problem occur after exercise?

8.Does the problem associated with fever?

9.Does the problem occur at rest?

10.Does the problem occur perennial

11.Does cough produce any sputum?

12.Sputum Color (Clear, Pink and Forthy, Thick yellow green, with boold or unknown)?

13.Do you have any swelling of feet and ankles?

Based on selection of these questionnaire the Expert System will be determined the disease Asthma or other diseases that occurs in Respiratory system such as Chronic Asthma, Congestive Heart Failure, Episodic Asthma, Respiratory Infection, Viral Infection, COPD, Cancer Association, unknown.

2 Algorithms

The extreme complexity of contemporary medical knowledge together with the intrinsic fallibility of human reasoning, have led to sustained efforts to develop clinical decision support systems, with the hope that bedside expert systems could overcome the limitations inherent to human cognition.

To further the research on computer-aided diagnosis begun in the 1960s, models of neural networks have been added to the pioneering work on artificial-intelligence Systems. The advent of artificial neural networks with the ability to identify multidimensional relationships in clinical data might improve the diagnostic power of the classical approaches. A great proportion of the neural network architectures applied to clinical diagnosis rests on multilayer feed-forward networks instructed with back propagation, followed by self-organizing maps and ART models. Although they perform with significant accuracy, this performance nevertheless remained insufficient to dispel the common fear that they are "black boxes" whose functioning cannot be well understood and, consequently, whose recommendations cannot be trusted.

The associative memory models, an early class of neural models that fit perfectly well with the vision of cognition emergent from today brain neuro-imaging techniques, are inspired on the capacity of human cognition to build semantic nets. Their known ability to support symbolic calculus makes them a possible link between connectionist models and classical artificial intelligence developments.

This work has three main objectives: a) to point out that associative memory models have the possibility to act as expert systems in medical diagnosis; b) to show in a simple and straightforward way how to instruct a minimal expert system with associative memories; and c) to encourage the implementation of this methodology at large scale by medical groups. Therefore, we address the building of associative memory-based expert systems for the medical diagnosis domain.

Also, we presented a comparative study with other machine learning algorithms such as Bayesian networks, ID3 and C4.5.

Models: We present the context-dependent auto-associative memory model, in addition to this, we present a brief about the remain algorithms. The sets of diseases and symptoms are mapped onto a pair of basis of orthogonal vectors. A matrix memory stores the associations between the signs and symptoms, and their corresponding diseases.

In order to provide a quick appreciation of the validity of the model and its potential clinical relevance we implemented an application with real data. A memory was trained with published data of Asthma data that could be represented as datasets from the questionnaire as well as clinical data. A set of personal observations used as a test set to evaluate the capacity of the model to discriminate between Asthma and asthma like diseases on the basis of questionnaire data as well as clinical and laboratory findings.

Auto-Associative Memory Neural networks: Also this algorithm is referred to as Context dependent auto-associative memory neural network, which is more powerful algorithm that suits to compute the clinical and laboratory factors effectively. Here, we could use the Kronecker product matrix as memory representation in the network structure. The model of this algorithm is presented in Figure1.

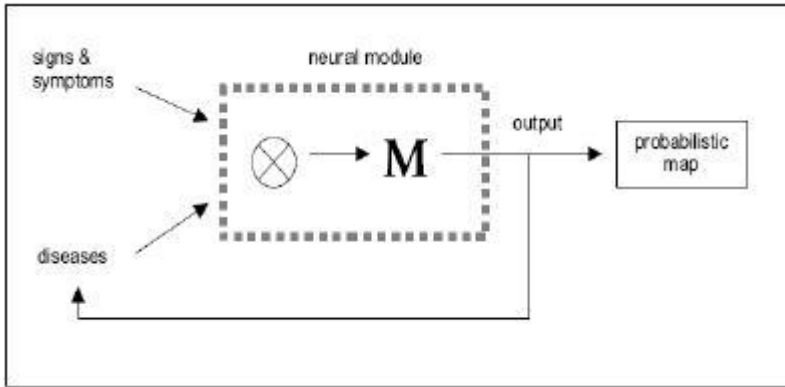


Fig. 1. AMNN model

The neural module receives the input of two vectors: one representing the set of possible diseases up to the moment and the other vector corresponding to a new sign, symptom or laboratory result. The action of the neurons that constitute the neural module can be divided into two steps: the Kronecker product of these two entries and the association of this stimulus with an output activity pattern. This output vector is a linear combination of a narrower set of disease vectors that can be reinjected if a new clinical data arrives or can be processed to obtain the probability attributable to each diagnostic decision.

A context-dependent associative memory M acting as a basic expert system is a matrix

$$M = \sum_{i=1}^k d_i [d_i \otimes \sum_{j(i)} s_j]^T \quad (1)$$

where d_i are column vectors mapping k different diseases (the set d is chosen orthonormal), and $s_{j(i)}$ are column vectors mapping signs or symptoms accompanying the i disease (also an orthonormal set). The sets of symptoms corresponding to each disease can overlap. The Kronecker product between two matrices A and B is another matrix defined by

$$A \otimes B = a(i, j) \cdot B \quad (2)$$

denoting that each scalar coefficient of matrix A , $a(i, j)$, is multiplied by the entire matrix B . Hence, if A is $n \times m$ dimensional and B is $k \times l$ dimensional, the resultant matrix will have the dimension $nk \times ml$.

Using this neural networks algorithm the memory representation is more powerful and effective mechanism to compute the probable disease based on symptoms and signs given by the user.

Decision Tree Algorithms. Decision tree algorithms are supervised machine learning algorithms that can be used to classify the diseases with the use of signs and symptoms of asthma. These algorithms represent the data in the form of datasets. We represented the signs and symptoms data as datasets that can supplied to these algorithms to predict the asthma or asthma like symptoms. Three file formats could be used for these algorithms.

1) **Asthma.names**, which represents the classification features of the disease, includes disease types and its symptoms defined from questionnaire and clinical reports.

2) **Asthma.data**, which represents the training data and

3) **Asthma.test**, represents test data.

The clinical reports may be either discrete or continuous data format. We considered the ID3 and C4.5 algorithms from decision trees. ID3 algorithm could not able to compute the continuous data effectively and also produce much noise, which gives less efficient than auto-associative neural networks. And then we applied C4.5 algorithm for predicting the disease effectively using asthma symptoms from questionnaire and clinical reports, which gives much efficient than ID3 algorithm. C4.5 is a statistical based decision tree algorithm that is from pruning of ID3 algorithm to reduce the noise as well as compute the continuous data. Similarly we applied Bayesian networks in addition to these algorithms, which is a pure probability based algorithm, that also computes the noise level and gives the exact disease that can be matched from the symptoms and signs.

3 Comparative Study

By applying these algorithms on the signs and symptoms of asthma and asthma like diseases, we analyzed and studied from various factors such as validity,

Table 2.

Accuracy	
Algorithm	Accuracy
Context Sensitive auto-associative memory neural networks(AMNN)	75.23 +/- 1.52
Bayesian Networks(BN)	72.54 +/- 1.67
ID3	70.13 +/- 1.68
C4.5	74.11 +/- 1.69

reliability, effectiveness and accuracy of proper outcomes that can be mapped with Experts knowledge. From these case studies, we determined the Context dependent auto-associative memory neural networks are one the best algorithm than Bayesian networks, ID3 and C4.5, which we have chosen from machine learning algorithms. Also, the results of expert system will be compared with the physical diagnosis in terms of sensitivity and specificity. The sensitivity was about 100patients reports. The accuracy of chosen algorithms will be presented below.

Also, the outstanding characteristics such as complexity, efficiency, accuracy and so on of these algorithms will be presented in the following table:

Table 3. Outstanding characteristics of different models

Characteristics	AMNN	BN	ID3	C4.5
Complexity	High	High	Moderate	Optimal
Efficiency	Good	Better	Low	Better
Accuracy	Very good	Good	Normal	Good
Nonlinear data	Good	Good	Low	Good
Explaining Capacity	Excellent	Good	Normal	Good
Narrowing diagnostic possibilities	Excellent	Good	Normal	Good
Automatic assignment of probabilities	Good	Good	Normal	Normal

4 Discussions and Conclusions

We have shown here that context-dependent associative memories could act as medical decision support systems. The system implies the previous coding of a set of diseases and its corresponding semi logic findings on individual basis of orthogonal vectors. The model presented in this communication is only a minimal module able to evaluate the probabilities of different diagnoses when a set of signs and symptoms is presented to it. This expert system based on an associative memory shares with programs using artificial intelligence a great capacity, quickly to narrow the number of diagnostic possibilities. Also, it is able to cope with variations in the way that a disease can present itself. A clear advantage of this system is that the probability assignment to the different diagnostic possibilities in any particular clinical situation does not have to be arbitrarily assigned by the specialist, but is automatically provided by the system, in agreement with the acquired experience.

The relevant properties of this associative memory model are summarized in Table 2 and 3 in comparison to other models.

We conclude that context-dependent associative memory model is a promising alternative in the development of accuracy diagnostic tools. We expect that its easy implementation stimulates groups of medical informatics to develop this expert system at real scale than other machine learning algorithms.

Acknowledgement

We thank Dr. K. Hari Prasad, Prof K.P.R Chowdary, Mr. Girija Shankar and Dr. S Kumar in the design and development of expert systems and also revision and improvements of such model.

References

- [1] Pomi, A., Olivera, F.: Context-sensitive auto associative memories as expert systems in medical diagnosis, PubMed publications (2006)
- [2] Torén, K., Brisman, J., Järholm, B.: Asthma and asthma-like symptoms in adults assessed by questionnaires. A literature review., *Chest*. 104, 600–608 (1993)
- [3] Mizraji, E., Pomi, A., Alvarez, F.: Multiplicative contexts in associative memories. *BioSystems* 32, 145–161 (1994)
- [4] Anderson, J.A., Cooper, L., Nass, M.M., Freiburger, W., Grenander, U.: Some properties of a neural model for memory. In: AAAS Symposium on Theoretical Biology and Biomathematics (1972)
- [5] Szolovits, P., Patil, R.S., Schwartz, W.B.: Artificial Intelligence in medical diagnosis. *Annals of Internal Medicine* 108, 80–87 (1988)
- [6] Schwartz, W.B., Patil, R.S., Szolovits, P.: Artificial Intelligence in medicine: Where do we stand? *New England Journal of Medicine* 316, 685–688 (1987)
- [7] UCI machine learning Repository for datasets representation,
<http://archive.ics.uci.edu/ml/machine-learning-databases/>,
<http://www.decisiontrees.net/node/>, <http://www.easydiagnosis.com>

Stabilization of Large Scale Linear Discrete-Time Systems by Reduced Order Controllers

Sundarapandian Vaidyanathan and Kavitha Madhavan

Department of Mathematics, Vel Tech Dr. RR & Dr. SR Technical University
Avadi-Alamathi Road, Avadi, Chennai-600 062, India
sundarvtu@gmail.com
<http://www.vel-tech.org/>

Abstract. This paper investigates the stabilization of large scale discrete-time linear systems by reduced order controllers. Conditions are derived for the design of reduced order controllers for the discrete-time linear systems by obtaining a reduced order model of the original plant using the dominant state of the system. The reduced order controllers are assumed to use only the state of the reduced order model of the original plant.

Keywords: Model reduction, reduced-order controllers, stabilization, dominant state, discrete-time linear systems.

1 Introduction

During the past four decades, a significant attention has been paid to the construction of reduced-order observers and stabilization using reduced-order controllers for linear control systems ([1]-[10]). Especially in recent decades, the control problem of large-scale linear systems has been an active area of research. This is due to practical and technical issues like information transfer networks, data acquisition, sensing, computing facilities and the associated cost involved which stem from using full order controller design. Thus, there is a great demand for the control of large scale linear systems with the use of reduced-order controllers rather than full-order controllers ([1]-[3]).

In this paper, we derive a reduced-order model for any linear discrete-time control system and our approach is based on the approach of using the *dominant* state of the given linear discrete-time control system, i.e. we derive the reduced-order model for a given linear discrete-time control system keeping only the dominant state of the given linear plant. The *dominant* state of a linear control system corresponds to the *slow modes* of the linear system, while the *non-dominant state* of the control system corresponds to the *fast modes* of the linear system ([3]-[10]).

As an application of our recent work ([9]-[10]), we first derive the reduced-order model of the given linear discrete-time control system. Using the reduced-order model obtained, we characterize the existence of a reduced-order controller that stabilizes the full linear system, using only the dominant state of the system.

This paper is organized as follows. In Section 2, we derive sufficient conditions for the derivation of reduced order model for the original large scale linear system and using this reduced order model, we derive conditions for the existence of reduced order

controllers for the original system that uses only the state of the reduced-order model. In Section 3, a numerical example is shown to verify the result. Conclusions are contained in the final section.

2 Reduced Order Controller Design for Linear Systems

Consider a large scale linear discrete-time system given by

$$x(k+1) = Ax(k) + Bu(k) \quad (1)$$

where $x \in \mathbb{R}^n$ is the *state* and $u \in \mathbb{R}^m$ is the *control input*.

We assume that A and B are constant matrices with real entries of dimensions $n \times n$ and $n \times m$ respectively.

First, we suppose that we have performed an identification of the *dominant (slow)* and *non-dominant (fast)* states of the original linear system (1) using the modal approach as described in [9].

Without loss of generality, we may assume that

$$x = \begin{bmatrix} x_s \\ x_f \end{bmatrix},$$

where $x_s \in \mathbb{R}^r$ represents the *dominant* state and $x_f \in \mathbb{R}^{n-r}$ represents the *non-dominant* state.

Then the system (1) takes the form

$$\begin{bmatrix} x_s(k+1) \\ x_f(k+1) \end{bmatrix} = \begin{bmatrix} A_{ss} & A_{sf} \\ A_{fs} & A_{ff} \end{bmatrix} \begin{bmatrix} x_s(k) \\ x_f(k) \end{bmatrix} + \begin{bmatrix} B_s \\ B_f \end{bmatrix} u(k) \quad (2)$$

i.e.

$$\begin{aligned} x_s(k+1) &= A_{ss} x_s(k) + A_{sf} x_f(k) + B_s u(k) \\ x_f(k+1) &= A_{fs} x_s(k) + A_{ff} x_f(k) + B_f u(k) \end{aligned} \quad (3)$$

For the sake of simplicity, we shall assume that the matrix A has distinct eigenvalues. We note that this condition is usually satisfied in most practical situations. Then it follows that A is diagonalizable.

Thus, we can find a nonsingular (modal) matrix M consisting of the n linearly independent eigenvectors of A so that the transformation

$$x(k) = Mz(k) \quad (4)$$

results in the original system (2) being transformed into the following diagonal form:

$$z(k+1) = \Lambda z(k) + \Gamma u(k) \quad (5)$$

where

$$\Lambda = M^{-1}AM = \begin{bmatrix} \Lambda_s & 0 \\ 0 & \Lambda_f \end{bmatrix} \quad (6)$$

is a diagonal matrix consisting of the n eigenvalues of A and

$$\Gamma = M^{-1}B = \begin{bmatrix} \Gamma_s \\ \Gamma_f \end{bmatrix} \quad (7)$$

Thus, the plant (5) can be written as

$$\begin{bmatrix} z_s(k+1) \\ z_f(k+1) \end{bmatrix} = \begin{bmatrix} \Lambda_s & 0 \\ 0 & \Lambda_f \end{bmatrix} \begin{bmatrix} z_s(k) \\ z_f(k) \end{bmatrix} + \begin{bmatrix} \Gamma_s \\ \Gamma_f \end{bmatrix} u(k) \quad (8)$$

i.e.

$$\begin{aligned} z_s(k+1) &= \Lambda_s z_s(k) + \Gamma_s u(k) \\ z_f(k+1) &= \Lambda_f z_f(k) + \Gamma_f u(k) \end{aligned} \quad (9)$$

Next, we make the following assumptions:

- (H1) As $k \rightarrow \infty$, $z_f(k+1) \approx z_f(k)$, i.e. z_f takes a constant value in the steady-state.
 (H2) The matrix $I - \Lambda_f$ is invertible.

Then it follows from (9) that for large values of k , we have

$$z_f(k) \approx \Lambda_f z_f(k) + \Gamma_f u(k) \quad (10)$$

i.e.

$$z_f(k) \approx (I - \Lambda_f)^{-1} \Gamma_f u(k) \quad (11)$$

The inverse of the transformation (4) is given by

$$z(k) = M^{-1} x(k) = G x(k) \quad (12)$$

i.e.

$$\begin{bmatrix} z_s(k) \\ z_f(k) \end{bmatrix} = \begin{bmatrix} G_{ss} & G_{sf} \\ G_{fs} & G_{ff} \end{bmatrix} \begin{bmatrix} x_s(k) \\ x_f(k) \end{bmatrix} \quad (13)$$

i.e.

$$\begin{aligned} z_s(k) &= G_{ss} x_s(k) + G_{sf} x_f(k) \\ z_f(k) &= G_{fs} x_s(k) + G_{ff} x_f(k) \end{aligned} \quad (14)$$

Next, we assume the following:

- (H3) The matrix G_{ff} is invertible.

Then the second equation of (14) can be rewritten as

$$x_f(k) = -G_{ff}^{-1} G_{fs} x_s(k) + G_{ff}^{-1} z_f(k) \quad (15)$$

Substituting (10) into (15), we get

$$x_f(k) = P x_s(k) + Q u(k), \quad (16)$$

where

$$P = -G_{ff}^{-1} G_{fs} \text{ and } Q = -G_{ff}^{-1} (I - A_f)^{-1} \Gamma_f \tag{17}$$

Substituting (16) into (3), we get the reduced order model of the original linear system as

$$x_s(k + 1) = A_s^* x_s(k) + B_s^* u(k) \tag{18}$$

where

$$A_s^* = A_{ss} + A_{sf}P \text{ and } B_s^* = B_s + A_{ff}Q \tag{19}$$

Thus, under the assumptions (H1)-(H3), the original linear system (1) can be expressed in a simplified form as

$$\begin{bmatrix} x_s(k + 1) \\ x_f(k + 1) \end{bmatrix} = \begin{bmatrix} A_s^* & 0 \\ A_f^* & 0 \end{bmatrix} \begin{bmatrix} x_s(k) \\ x_f(k) \end{bmatrix} + \begin{bmatrix} B_s^* \\ B_f^* \end{bmatrix} u(k) \tag{20}$$

In this paper, we consider the design of reduced order controller for the linear system (20) using only the dominant state x_s of the system. Thus, for the linear system (20), we study the problem of finding a feedback controller of the form

$$u(k) = [F_s^* \quad 0] \begin{bmatrix} x_s(k) \\ x_f(k) \end{bmatrix} = F_s^* x_s(k) \tag{21}$$

so that the resulting closed-loop system governed by the equations

$$\begin{aligned} x_s(k + 1) &= (A_s^* + B_s^* F_s^*) x_s(k) \\ x_f(k + 1) &= (A_f^* + B_f^* F_s^*) x_s(k) \end{aligned} \tag{22}$$

is exponentially stable. [Note that the stabilizing feedback control law (21), if it exists, will also stabilize the reduced-order linear system (18).]

From the first equation of (22), it follows that

$$x_s(k) = (A_s^* + B_s^* F_s^*)^k x_s(0) \tag{23}$$

which shows that $x_s(k) \rightarrow 0$ as $k \rightarrow \infty$ if and only if $A_s^* + B_s^* F_s^*$ is a convergent matrix, *i.e.* it has all eigenvalues inside the unit circle of the complex plane.

From the second equation of (22), it follows that if $x_s(k) \rightarrow 0$, then $x_f(k) \rightarrow 0$ as $k \rightarrow \infty$. Thus, for the stability analysis of the closed-loop system (22), it suffices to consider only the first equation of (22). Hence, the problem reduces to finding stabilizing controllers for the reduced-order linear system (18).

Thus, we conclude that the reduced order controller for the linear system (20) is solvable if and only if the system pair (A_s^*, B_s^*) is stabilizable, *i.e.* there exists a gain matrix F_s^* such that $A_s^* + B_s^* F_s^*$ is convergent. Especially, if the system pair

$$(A_s^*, B_s^*)$$

is completely controllable, then the eigenvalues of the closed-loop system matrix

$$A_s^* + B_s^* F_s^*$$

can be arbitrarily placed in the complex plane. In particular, we can always find a gain matrix F_s^* such that the closed-loop system matrix $A_s^* + B_s^* F_s^*$ is convergent.

Hence, we obtain the following result.

Theorem 1. *Suppose that the assumptions (H1)-(H3) hold. Then the system (18) is a reduced-order model for the original linear system (1). Also, the original linear system (1) can be expressed by the equations (20). Next, the feedback control law (21) that uses only the dominant state of the system stabilizes the full linear system (20) if and only if it stabilizes the reduced order linear system (18). Thus, the reduced order feedback controller problem is solvable if and only if the system pair (A_s^*, B_s^*) of the reduced-order linear system (18) is stabilizable, i.e. there exists a feedback gain matrix F_s^* so that the closed-loop system matrix $A_s^* + B_s^* F_s^*$ is convergent. If the system pair (A_s^*, B_s^*) of the reduced-order linear system (18) is completely controllable, then we can always find a feedback gain matrix F_s^* such that $A_s^* + B_s^* F_s^*$ is convergent with any desired eigenvalues inside the unit circle of the complex plane and hence we can construct the feedback control law (21) that stabilizes the full linear system (20). ■*

3 A Numerical Example

Consider the fourth order linear discrete-time control system described by

$$x(k + 1) = Ax(k) + Bu(k) \tag{24}$$

where

$$A = \begin{bmatrix} 2.9845 & -0.2156 & -0.0123 & 0.0023 \\ 0.8267 & 1.8934 & -0.0312 & 0.0002 \\ -0.0125 & 0.0352 & 0.0024 & 0.0235 \\ -0.0012 & 0.0022 & -0.0558 & 0.2041 \end{bmatrix} \text{ and } B = \begin{bmatrix} 0.4214 \\ 0.5378 \\ 0.4476 \\ 0.3425 \end{bmatrix} \tag{25}$$

The eigenvalues of the matrix A are

$$\lambda_1 = 2.7844, \lambda_2 = 2.0930, \lambda_3 = 0.1974, \lambda_4 = 0.0096$$

Thus, we note that λ_1, λ_2 are unstable (slow) eigenvalues and λ_3, λ_4 are stable (fast) eigenvalues for the system matrix A .

The dominance measure of the eigenvalues is calculated as in [9] and obtained as

$$\Omega = \begin{bmatrix} 0.2101 & 0.0406 & 0.0000 & 0.0021 \\ 0.1949 & 0.1680 & 0.0006 & 0.0060 \\ 0.0015 & 0.0026 & 0.0337 & 0.4191 \\ 0.0000 & 0.0001 & 0.2783 & 0.1202 \end{bmatrix}$$

To determine the dominance of the k th eigenvalue in all the n states, we use the measure (see [9])

$$\Theta_k = \sum_{i=1}^n \Omega_{ik}$$

Thus, we obtain

$$\Theta = [0.4065 \quad 0.2113 \quad 0.3126 \quad 0.5473]$$

To determine the relative dominance of the k th eigenvalue in the i th state, we use the measure (see [9])

$$\phi_{ik} = \left| \frac{\Omega_{ik}}{\Theta_k} \right| \times 100$$

Thus, we obtain

$$\Phi = \begin{bmatrix} 51.69 & 19.21 & 0 & 0.38 \\ 47.95 & 79.51 & 0.19 & 1.1 \\ 0.37 & 1.23 & 10.78 & 76.57 \\ 0 & 0.05 & 89.03 & 37.29 \end{bmatrix}$$

The relative contribution of $\zeta = 2$ eigenvalues in the i th state is determined from

$$\psi_i = \sum_{k=1}^{\zeta} \phi_{ik} \quad (i = 1, 2, 3, 4)$$

which gives

$$\Psi = [70.9 \quad 127.46 \quad 1.6 \quad 0.05]$$

Thus, it follows that the first two states (x_1, x_2) are dominant (slow) states, while the last two states (x_3, x_4) are the non-dominant (fast) states of the linear system [24].

Using the procedure described in Section 2, the reduced-order linear model for the given linear system [24] is obtained as

$$x_s(k + 1) = A_s^* x_s(k) + B_s^* u(k) \tag{26}$$

where

$$x_s = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad A_s^* = \begin{bmatrix} 2.9846 & -0.2158 \\ 0.8270 & 1.8929 \end{bmatrix}, \quad B_s^* = \begin{bmatrix} 0.4261 \\ 0.5518 \end{bmatrix} \tag{27}$$

The matrix A_s^* has two unstable eigenvalues 2.7844, 2.0930, which coincide with the unstable (slow) eigenvalues of A .

Since the system pair (A_s^*, B_s^*) of the reduced-order system [26] is completely controllable, the reduced order controller problem for the given linear system is solvable by Theorem 1 and a stabilizing feedback control law for the reduced order linear system [26] is obtained as

$$u = F_s^* x_s = [-69.6826 \quad 45.3321] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{28}$$

Upon substitution of the control law [28] into the reduced-order linear system [26], we obtain the closed-loop system

$$x_s(k + 1) = (A_s^* + B_s^* F_s^*) x_s(k) \tag{29}$$

which has the stable eigenvalues $\lambda_1^* = 0.1$ and $\lambda_2^* = 0.1$.

Thus, the closed-loop system (29) is globally exponentially stable.

The response $x_s(k)$ of the closed-loop system (29) for the initial state

$$x_s(0) = \begin{bmatrix} 100 \\ 100 \end{bmatrix}$$

is shown in Figure 1. Note that the response $x_s(k)$ converges to zero in about 5 seconds.

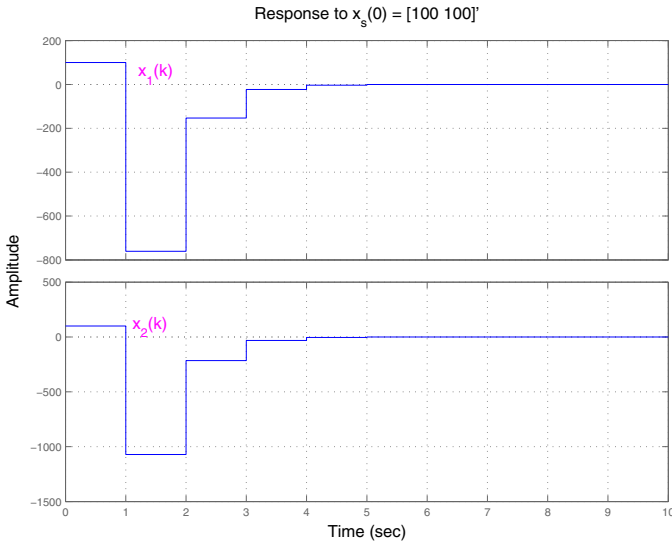


Fig. 1. Time Responses of the Closed-Loop System (29)

4 Conclusions

In this paper, sufficient conditions are derived for the design of reduced order controllers by obtaining a reduced order model of the original plant using the dominant state of the system. The reduced order controllers are assumed to use only the state of the reduced order model of the original plant. An example has been given to illustrate the derivation of the reduced order model and the design of reduced order controller for a four-dimensional linear discrete-time system.

References

1. Cumming, S.D.: Design of observers of reduced dynamics. *Electronics Letters* 5, 213–214 (1969)
2. Fortman, T.E., Williamson, D.: Design of low-order observers for linear feedback control laws. *IEEE Trans. Automatic Control*. 17, 301–308 (1971)

3. Litz, L., Roth, H.: State decomposition for singular perturbation order reduction – a modal approach. *Internat. J. Control.* 34, 937–954 (1981)
4. Lastman, G., Sinha, N., Rozsa, P.: On the selection of states to be retained in reduced-order model. *IEE Proc., Part D.* 131, 15–22 (1984)
5. Anderson, B.D.O., Liu, Y.: Controller reduction: concepts and approaches. *IEEE Trans. Automatic Control.* 34, 802–812 (1989)
6. Mustafa, D., Glover, K.: Controller reduction by H_∞ balanced truncation. *IEEE Trans. Automatic Control* 36, 668–682 (1991)
7. Aldeen, M.: Interaction modelling approach to distributed control with application to interconnected dynamical systems. *Internat. J. Control.* 53, 1035–1054 (1991)
8. Aldeen, M., Trinh, H.: Observing a subset of the states of linear systems. *IEE Proc. Control Theory Appl.* 141, 137–144 (1994)
9. Sundarapandian, V.: Distributed control schemes for large scale interconnected discrete-time linear systems. *Math. Computer Modelling.* 41, 313–319 (2005)
10. Sundarapandian, V., Kavitha, M., Ravichandran, C.S.: Reduced order model using the dominant state of linear discrete-time systems. *Internat. J. Computational Applied Math.* 5, 301–312 (2010)

Hybrid Synchronization of Hyperchaotic Qi and Lü Systems by Nonlinear Control

Sundarapandian Vaidyanathan and Suresh Rasappan

Department of Mathematics, Vel Tech Dr. RR & Dr. SR Technical University
Avadi-Alamathi Road, Avadi, Chennai-600 062, India
sundarvtu@gmail.com
<http://www.vel-tech.org/>

Abstract. This paper investigates the hybrid synchronization of identical hyperbolic Qi systems and hybrid synchronization of hyperchaotic Qi and Lü systems. The hyperchaotic Qi system (2008) and hyperchaotic Lü system (2006) are important models of hyperchaotic systems. Hybrid synchronization of the hyperchaotic systems is achieved through synchronization of two pairs of states and anti-synchronization of the other two pairs of states of the two hyperchaotic systems. Nonlinear control is the method used for the hybrid synchronization of identical hyperbolic Qi systems and hybrid synchronization of hyperchaotic Qi and Lü systems. Since the Lyapunov exponents are not required for these calculations, this method is effective and convenient to achieve hybrid synchronization of the two hyperchaotic systems. Numerical simulations are shown to verify the results.

Keywords: Hybrid synchronization, hyperchaos, hyperchaotic Qi system, hyperchaotic L system, nonlinear control.

1 Introduction

Chaotic systems are dynamical systems that are highly sensitive to initial conditions. This sensitivity is popularly referred to as the butterfly effect [1]. Chaos is an interesting nonlinear phenomenon and has been extensively and intensively studied in the last two decades ([1]-[23]). In 1990, Pecora and Carroll [2] introduced a method to synchronize two identical chaotic systems and showed that it was possible for some chaotic systems to be completely synchronized. From then on, chaos synchronization has been widely explored in a variety of fields including physical [3], chemical [4], ecological [5] systems, secure communications ([6]-[7]) etc.

In most of the chaos synchronization approaches, the master-slave or drive-response formalism is used. If a particular chaotic system is called the *master* or *drive system* and another chaotic system is called the *slave* or *response system*, then the idea of synchronization is to use the output of the master system to control the slave system so that the output of the slave system tracks the output of the master system asymptotically.

Since the seminal work by Pecora and Carroll [2], a variety of impressive approaches have been proposed for the synchronization of the chaotic systems such as the PC

method [2], sampled-data feedback synchronization method [8], OGY method [9], time-delay feedback method [10], backstepping method [11], adaptive design method [12], sliding mode control method [13], etc.

So far, many types of synchronization phenomenon have been presented such as complete synchronization [2], phase synchronization ([5], [14]), generalized synchronization ([7], [15]), anti-synchronization ([16], [17]), projective synchronization [18], generalized projective synchronization ([19], [20]) etc.

Complete synchronization (CS) is characterized by the equality of state variables evolving in time, while anti-synchronization (AS) is characterized by the disappearance of the sum of relevant state variables evolving in time. Projective synchronization (PS) is characterized by the fact the master and slave systems could be synchronized up to a scaling factor, whereas in generalized projective synchronization (GPS), the responses of the synchronized dynamical states synchronize up to a constant scaling matrix α . It is easy to see that the complete synchronization and anti-synchronization are the special cases of the generalized projective synchronization where the scaling matrix $\alpha = I$ and $\alpha = -I$, respectively.

In hybrid synchronization of chaotic systems [20], one part of the systems is synchronized and the other part is anti-synchronized so that complete synchronization (CS) and anti-synchronization (AS) co-exist in the systems. The co-existence of CS and AS is very useful in secure communication and chaotic encryption schemes.

This paper is organized as follows. In Section 2, we derive results for the hybrid synchronization between identical hyperchaotic Qi systems ([22], 2008). In Section 3, we study the hybrid synchronization between hyperchaotic Qi system and hyperchaotic Lü system ([23], 2006). The nonlinear controllers are derived for hybrid synchronization of the two hyperchaotic systems. The nonlinear control method is simple, effective and easy to implement in practical applications. Conclusions are contained in the final section.

2 Hybrid Synchronization of Identical Qi Systems

In this section, we consider the hybrid synchronization of identical Qi systems [22]. Thus, we consider the *master* system as the hyperchaotic Qi dynamics described by

$$\begin{aligned}\dot{x}_1 &= a(x_2 - x_1) + x_2x_3 \\ \dot{x}_2 &= b(x_1 + x_2) - x_1x_3 \\ \dot{x}_3 &= -cx_3 - \epsilon x_4 + x_1x_2 \\ \dot{x}_4 &= -dx_4 + fx_3 + x_1x_2\end{aligned}\tag{1}$$

where $x_i (i = 1, 2, 3, 4)$ are the *state* variables and a, b, c, d, ϵ, f are positive constants.

When $a = 50, b = 24, c = 13, d = 8, \epsilon = 33$ and $f = 30$, the Qi system (1) is hyperchaotic (see Figure 1).

We consider the hyperchaotic Qi dynamics also as the *slave* system, which is described by the dynamics

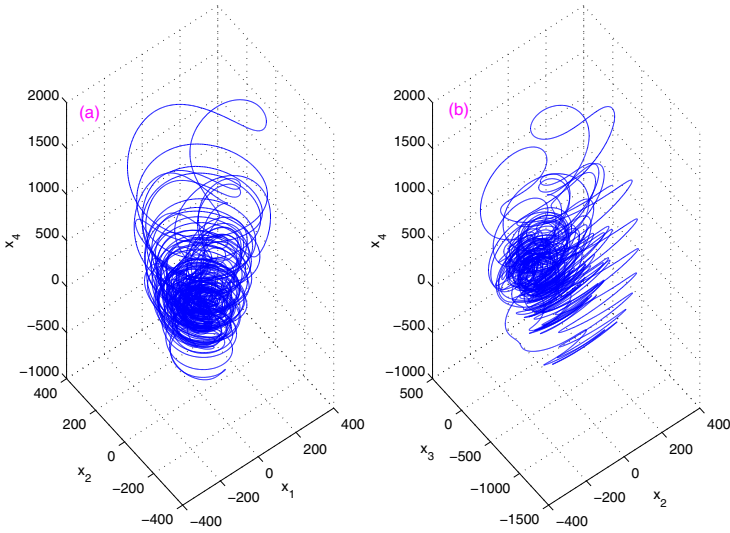


Fig. 1. State Orbits of the Hyperchaotic Qi System (1)

$$\begin{aligned}
 \dot{y}_1 &= a(y_2 - y_1) + y_2y_3 + u_1 \\
 \dot{y}_2 &= b(y_1 + y_2) - y_1y_3 + u_2 \\
 \dot{y}_3 &= -cy_3 - \epsilon y_4 + y_1y_2 + u_3 \\
 \dot{y}_4 &= -dy_4 + f y_3 + y_1y_2 + u_4
 \end{aligned}
 \tag{2}$$

where $y_i (i = 1, 2, 3, 4)$ are the *state* variables and $u_i (i = 1, 2, 3, 4)$ are the active controls.

For the hybrid synchronization of the identical hyperchaotic Qi systems (1) and (2), the *errors* are defined as

$$\begin{aligned}
 e_1 &= y_1 - x_1 \\
 e_2 &= y_2 + x_2 \\
 e_3 &= y_3 - x_3 \\
 e_4 &= y_4 + x_4
 \end{aligned}
 \tag{3}$$

From the error equations (3), it is obvious that one part of the two hyperchaotic system is completely synchronized (first and third states), while the other part is completely anti-synchronized (second and fourth states) so that complete synchronization (CS) and anti-synchronization (AS) co-exist in the synchronization process of the two hyperchaotic systems (1) and (2).

A simple calculation yields the error dynamics as

$$\begin{aligned}
 \dot{e}_1 &= -ae_1 + a(y_2 - x_2) + y_2y_3 - x_2x_3 + u_1 \\
 \dot{e}_2 &= be_2 + b(y_1 + x_1) - (y_1y_3 + x_1x_3) + u_2 \\
 \dot{e}_3 &= -ce_3 - \epsilon(y_4 - x_4) + y_1y_2 - x_1x_2 + u_3 \\
 \dot{e}_4 &= -de_4 + f(y_3 + x_3) + y_1y_2 + x_1x_2 + u_4
 \end{aligned}
 \tag{4}$$

We consider the nonlinear controller defined by

$$\begin{aligned}
 u_1 &= -a(y_2 - x_2) - y_2y_3 + x_2x_3 \\
 u_2 &= -2be_2 - b(y_1 + x_1) + y_1y_3 + x_1x_3 \\
 u_3 &= \epsilon(y_4 - x_4) - y_1y_2 + x_1x_2 \\
 u_4 &= -f(y_3 + x_3) - (y_1y_2 + x_1x_2)
 \end{aligned}
 \tag{5}$$

Substitution of (5) into (4) yields the linear error dynamics

$$\begin{aligned}
 \dot{e}_1 &= -ae_1 \\
 \dot{e}_2 &= -be_2 \\
 \dot{e}_3 &= -ce_3 \\
 \dot{e}_4 &= -de_4
 \end{aligned}
 \tag{6}$$

We consider the candidate Lyapunov function defined by

$$V(e) = \frac{1}{2} e^T e = \frac{1}{2} (e_1^2 + e_2^2 + e_3^2 + e_4^2)
 \tag{7}$$

Differentiating (7) along the trajectories of the system (6), we get

$$\dot{V}(e) = -ae_1^2 - be_2^2 - ce_3^2 - de_4^2
 \tag{8}$$

which is a negative definite function on \mathbb{R}^4 , since a, b, c, d are positive constants.

Thus, by Lyapunov stability theory [24], the error dynamics (6) is globally exponentially stable. Hence, we obtain the following result.

Theorem 1. *The identical hyperchaotic Qi systems (1) and (2) are globally and exponentially hybrid synchronized with the active nonlinear controller (5).* ■

Numerical Simulations For the numerical simulations, the fourth order Runge-Kutta method is used to solve the two systems of differential equations (1) and (2) with the nonlinear controller (5).

The parameters of the identical hyperchaotic Qi systems (1) and (2) are selected as

$$a = 50, b = 24, c = 13, d = 8, \epsilon = 33, f = 30$$

so that the systems (1) and (2) exhibit hyperchaotic behaviour.

The initial values for the master system (1) are taken as

$$x_1(0) = 6, x_2(0) = 3, x_3(0) = 2, x_4(0) = 4$$

and the initial values for the slave system (2) are taken as

$$y_1(0) = 1, y_2(0) = 4, y_3(0) = 3, y_4(0) = 5$$

Figure 2 shows the time response of the error states e_1, e_2, e_3, e_4 of the error dynamical system (4) when the active nonlinear controller (5) is deployed. From this figure, it is clear that all the error states decay to zero exponentially in about 0.6 sec and thus, both complete synchronization (CS) and anti-synchronization (AS) co-exist for the identical Qi systems (1) and (2).

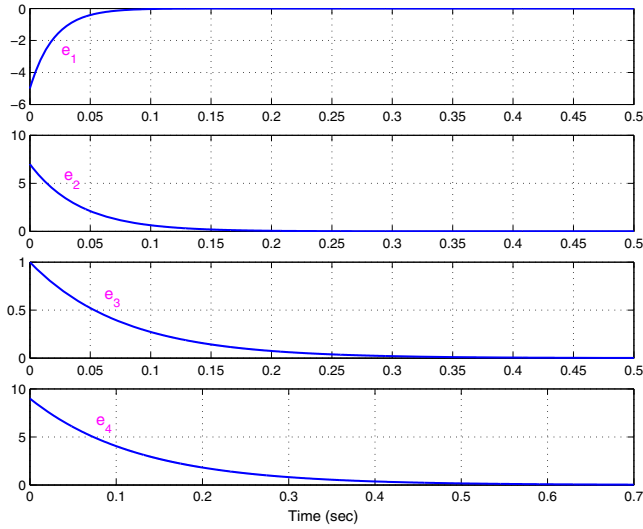


Fig. 2. Time Responses of the Error States of (4)

3 Hybrid Synchronization of Qi and Lü Systems

In this section, we consider the hyperchaotic Lü system [23] as the *master* system, which is described by the dynamics

$$\begin{aligned}
 \dot{x}_1 &= \alpha(x_2 - x_1) + x_4 \\
 \dot{x}_2 &= -x_1x_3 + \gamma x_2 \\
 \dot{x}_3 &= x_1x_2 - \beta x_3 \\
 \dot{x}_4 &= x_1x_3 + rx_4
 \end{aligned}
 \tag{9}$$

where $x_i (i = 1, 2, 3, 4)$ are the *state* variables and α, β, γ, r are positive constants.

When $\alpha = 36, \beta = 3, \gamma = 20, -0.35 \leq r \leq 1.3$, the hyperchaotic Lü system (9) is hyperchaotic (see Figure 3).

Next, we consider the hyperchaotic Qi dynamics [22] as the *slave* system, which is described by the dynamics

$$\begin{aligned}
 \dot{y}_1 &= a(y_2 - y_1) + y_2y_3 + u_1 \\
 \dot{y}_2 &= b(y_1 + y_2) - y_1y_3 + u_2 \\
 \dot{y}_3 &= -cy_3 - \epsilon y_4 + y_1y_2 + u_3 \\
 \dot{y}_4 &= -dy_4 + fy_3 + y_1y_2 + u_4
 \end{aligned}
 \tag{10}$$

where $y_i (i = 1, 2, 3, 4)$ are the *state* variables and $u_i (i = 1, 2, 3, 4)$ are the active controls.

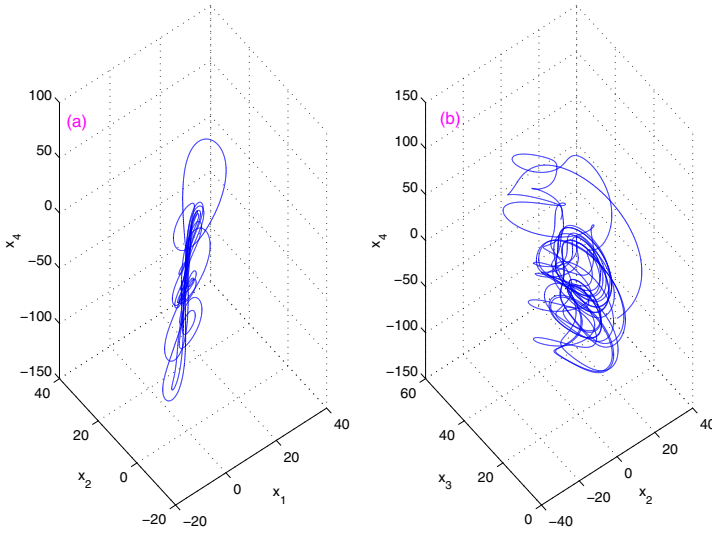


Fig. 3. State Orbits of the Hyperchaotic Lü System (9)

For the hybrid synchronization of the hyperchaotic Lü system (9) and Qi system (10), the errors are defined as

$$\begin{aligned}
 e_1 &= y_1 - x_1 \\
 e_2 &= y_2 + x_2 \\
 e_3 &= y_3 - x_3 \\
 e_4 &= y_4 + x_4
 \end{aligned}
 \tag{11}$$

From the error equations (11), it is obvious that one part of the two hyperchaotic system is completely synchronized (first and third states), while the other part is completely anti-synchronized (second and fourth states) so that complete synchronization (CS) and anti-synchronization (AS) co-exist in the synchronization process of the two hyperchaotic systems (9) and (10).

A simple calculation yields the error dynamics as

$$\begin{aligned}
 \dot{e}_1 &= -ae_1 + ay_2 + (\alpha - a)x_1 - \alpha x_2 - x_4 + y_2y_3 + u_1 \\
 \dot{e}_2 &= be_2 + by_1 + (\gamma - b)x_2 - (y_1y_3 + x_1x_3) + u_2 \\
 \dot{e}_3 &= -ce_3 - \epsilon y_4 + (\beta - c)x_3 + y_1y_2 - x_1x_2 + u_3 \\
 \dot{e}_4 &= -de_4 + fy_3 + (r + d)x_4 + y_1y_2 + x_1x_3 + u_4
 \end{aligned}
 \tag{12}$$

We consider the nonlinear controller defined by

$$\begin{aligned}
 u_1 &= -ay_2 - (\alpha - a)x_1 + \alpha x_2 + x_4 - y_2y_3 \\
 u_2 &= -2be_2 - by_1 - (\gamma - b)x_2 + y_1y_3 + x_1x_3 \\
 u_3 &= \epsilon y_4 - (\beta - c)x_3 - y_1y_2 + x_1x_2 \\
 u_4 &= -fy_3 - (r + d)x_4 - (y_1y_2 + x_1x_3)
 \end{aligned}
 \tag{13}$$

Substitution of (13) into (12) yields the linear error dynamics

$$\begin{aligned} \dot{e}_1 &= -ae_1 \\ \dot{e}_2 &= -be_2 \\ \dot{e}_3 &= -ce_3 \\ \dot{e}_4 &= -de_4 \end{aligned} \tag{14}$$

We consider the candidate Lyapunov function defined by

$$V(e) = \frac{1}{2} e^T e = \frac{1}{2} (e_1^2 + e_2^2 + e_3^2 + e_4^2) \tag{15}$$

Differentiating (15) along the trajectories of the system (14), we get

$$\dot{V}(e) = -ae_1^2 - be_2^2 - ce_3^2 - de_4^2 \tag{16}$$

which is a negative definite function on \mathbb{R}^4 , since a, b, c, d are positive constants.

Thus, by Lyapunov stability theory [24], the error dynamics (14) is globally exponentially stable. Hence, we obtain the following result.

Theorem 2. *The hyperchaotic Lü system (9) and the hyperchaotic Qi system (10) are globally and exponentially hybrid synchronized with the active nonlinear controller (13).* ■

Numerical Simulations For the numerical simulations, the fourth order Runge-Kutta method is used to solve the two systems of differential equations (9) and (10) with the nonlinear controller (13).

The parameters of the hyperchaotic Lü system (9) are selected as

$$\alpha = 36, \beta = 3, \gamma = 20, r = 1$$

and the parameters of the hyperchaotic Qi system (10) are selected as

$$a = 50, b = 24, c = 13, d = 8, \epsilon = 33, f = 30$$

so that the systems (9) and (10) exhibit hyperchaotic behaviour.

The initial values for the master system (9) are taken as

$$x_1(0) = 2, x_2(0) = 4, x_3(0) = 5, x_4(0) = 7$$

and the initial values for the slave system (2) are taken as

$$y_1(0) = 5, y_2(0) = 2, y_3(0) = 7, y_4(0) = 3$$

Figure 4 shows the time response of the error states e_1, e_2, e_3, e_4 of the error dynamical system (12) when the active nonlinear controller (13) is deployed. From this figure, it is clear that all the error states decay to zero exponentially in about 0.6 sec and thus, both complete synchronization (CS) and anti-synchronization (AS) co-exist for the hyperchaotic Lü system (9) and the hyperchaotic Qu system (10).

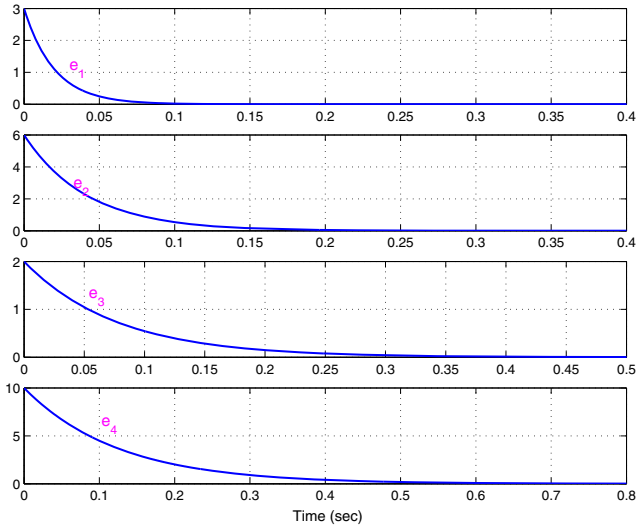


Fig. 4. Time Responses of the Error States of (12)

4 Conclusions

In this paper, nonlinear control method based on Lyapunov stability theory has been deployed to globally and exponentially hybrid synchronize two identical hyperbolic Qi systems and two different hyperchaotic systems, viz. hyperchaotic Qi system (2008) and hyperchaotic Lü system (2006). Numerical simulations are also shown to verify the proposed active nonlinear controllers to achieve hybrid synchronization of the two hyperchaotic systems. Since Lyapunov exponents are not required for the calculations, the proposed nonlinear control method is effective and convenient to achieve hybrid synchronization of the identical hyperchaotic Qi systems and the hybrid synchronization of hyperchaotic Qi and Lü systems.

References

1. Alligood, K.T., Sauer, T., Yorke, J.A.: Chaos: An Introduction to Dynamical Systems. Springer, New York (1997)
2. Pecora, L.M., Carroll, T.L.: Synchronization in chaotic systems. Phys. Rev. Lett. 64, 821–824 (1990)
3. Lakshmanan, M., Murali, K.: Chaos in Nonlinear Oscillators: Controlling and Synchronization. World Scientific, Singapore (1996)
4. Han, S.K., Kerrer, C., Kuramoto, Y.: Dephasing and bursting in coupled neural oscillators. Phys. Rev. Lett. 75, 3190–3193 (1995)
5. Blasius, B., Huppert, A., Stone, L.: Complex dynamics and phase synchronization in spatially extended ecological system. Nature 399, 354–359 (1999)
6. Feki, M.: An adaptive chaos synchronization scheme applied to secure communication. Chaos, Solit. Fract. 18, 141–148 (2003)

7. Murali, K., Lakshmanan, M.: Secure communication using a compound signal from generalized synchronizable chaotic systems. *Phys. Rev. Lett. A* 241, 303–310 (1998)
8. Yang, T., Chua, L.O.: Control of chaos using sampled-data feedback control. *Internat. J. Bifurcat. Chaos.* 9, 215–219 (1999)
9. Ott, E., Grebogi, C., Yorke, J.A.: Controlling chaos. *Phys. Rev. Lett.* 64, 1196–1199 (1990)
10. Park, J.H., Kwon, O.M.: A novel criterion for delayed feedback control of time-delay chaotic systems. *Chaos, Solit. Fract.* 17, 709–716 (2003)
11. Yu, Y.G., Zhang, S.C.: Adaptive backstepping synchronization of uncertain chaotic systems. *Chaos, Solit. Fract.* 27, 1369–1375 (2006)
12. Liao, T.L., Tsai, S.H.: Adaptive synchronization of chaotic systems and its applications to secure communications. *Chaos, Solit. Fract.* 11, 1387–1396 (2000)
13. Konishi, K., Hirai, M., Kokame, H.: Sliding mode control for a class of chaotic systems. *Phys. Lett. A* 245, 511–517 (1998)
14. Ge, Z.M., Chen, C.C.: Phase synchronization of coupled chaotic multiple time scales systems. *Chaos, Solit. Fract.* 20, 639–647 (2004)
15. Wang, Y.W., Guan, Z.H.: Generalized synchronization of continuous chaotic systems. *Chaos, Solit. Fract.* 27, 97–101 (2006)
16. Zhang, X., Zhu, H.: Anti-synchronization of two different hyperchaotic systems via active and adaptive control. *Inter. J. Nonlinear Science* 6, 216–223 (2008)
17. Chiang, T., Lin, J., Liao, T., Yan, J.: Anti-synchronization of uncertain unified chaotic systems with dead-zone nonlinearity. *Nonlinear Anal.* 68, 2629–2637 (2008)
18. Qiang, J.: Projective synchronization of a new hyperchaotic Lorenz system. *Phys. Lett. A* 370, 40–45 (2007)
19. Jian-Ping, Y., Chang-Pin, L.: Generalized projective synchronization for the chaotic Lorenz system and the chaotic Chen system. *J. Shanghai Univ.* 10, 299–304 (2006)
20. Li, R.H., Xu, W., Li, S.: Adaptive generalized projective synchronization in different chaotic systems based on parameter identification. *Phys. Lett. A* 367, 199–206 (2007)
21. Li, R.-h.: A special full-state hybrid projective synchronization in symmetrical chaotic systems. *Applied Math. Comput.* 200, 321–329, (2008)
22. Qi, G., Wyk, M.A., Wyk, B.J., Chen, G.: On a new hyperchaotic system. *Phys. Lett. A* 372, 124–136 (2008)
23. Chen, A., Lü, J., Lu, J., Yu, S.: Generating hyperchaotic Lü attractor via state feedback control. *Physica A* 364, 103–110 (2006)
24. Hahn, W.: *The Stability of Motion*. Springer, New York (1967)

An Intelligent Automatic Story Generation System by Revising Proppian's System

Jaya A.¹ and Uma G.V.²

¹ Research Schola

² Professor

Department of Computer Science Engineering,

Anna University, Chennai – 600 025, India

jaya_venky@yahoo.com, gvuma@annauniv.edu

Abstract. In the artificial intelligence era, there are so many story generators available for generating stories automatically. Story generators adapt either static theme conception or random theme conception procedure to conceive a theme. Similarly, most of the story generators use random sequence of predefined sentences to represent the story. This paper aims to construct a new story generator by revising the existing proppians story generator that generates the story automatically using 31 move functions. But the proppians story generator lacks in conceptual presentation and semantic reasoning. RPG (Revised Proppian story Generator) contains five story move functions and also each generated statement in the story undergoing the process of reasoning in order to check the semantic. Ontology supports this story generation system by providing nouns, related verbs, attributes, themes for generating a simple story. Moreover it helps to preserve the semantic of the story generated. Ontology is a formal explicit specification of shared conceptualization. This intelligent automatic story generation system helps for users to construct new meaningful stories using ontology in an efficient way.

Keywords: Story generation, ontology, Reasoning, Proppian system, Knowledge representation.

1 Introduction

Stories are naturally has their own way of attraction from children to old age people. Children learn their moral and social obligations in the form of stories narrated to them by their guardians and peers [1]. The basic characteristics of human beings can be crafted by narrating stories to the youngsters to inspire them. Hence, they play a vital role in everyone's life and their traits can be shaped well, based on the characters in the story. For example, Mahatma Gandhiji was influenced by the story "Harichandra" for speaking the truth in any kind of difficulties, and the "shravana story" for obedience to his parents. The story is a natural verbal description of objects / human beings, their attributes, relationships, beliefs, behaviors, motivations and interaction. It is a message that tells the particulars of an act, and an occurrence or course of events are presented in the form of writing. Writing story is an art which needs creativity and imagination to

represent the conceived theme. Some people may be fond of writing stories others may not be. This RPG enables the user to create small stories based on their desires. Each story adapts the plan or structure for story generation. It should have a proper structure to describe the settings, locations, characters to be played, and the sequence of events utilized to lead the story. Mark Riedl and Carlos León [9] proposed the following plan for a story.

A **domain** $D = \langle S, A \rangle$ is a tuple such that S is a description of the state of a world and A is the set of all possible operations that can change the world. A **narrative plan** $p = \langle I, A, O \rangle$ is a tuple such that I is a description of the initial state, A is a set of ground operators – called actions – and O is a set of temporal ordering constraints of the form $a1 < a2$ where $a1, a2 \in A$, and $a1$ necessarily precede $a2$ in the story.

The story plan operates as a tuple $\langle h, P, E \rangle$, where: h is the head of the action, defining its name and its arguments. For example, consider the function, “eat (*Lion, meat*)”; ‘eat’ is an action performed by the ground arguments *Lion* and *meat*, showing that “*Lion ate the meat*”. P is the set of ground preconditions, a set of propositions that define the previous state needed for the action in the story to be performed. E is the set of ground effects, that is, the set of propositions that are made true when the action is performed. For example, after the action “eat (*Lion, meat*)” is applied, the new state would contain the proposition *has (Lion, meat)*. This narrative plan helps to adhere the story generation in an efficient way. Story generators adapt either static theme conception or random theme conception procedure to conceive a theme. The simple grammar for short stories is described below:

Story \rightarrow Setting + Theme + Resolution
 Setting \rightarrow Characters + Location + Time
 Theme \rightarrow (Event)* + Goal
 Resolution \rightarrow Event State

By utilizing, conceived theme, settings, predefined functions and locations, a simple story can be constructed. Section 2 describes the existing story generators and reasoning methodologies and their strengths and weaknesses. Section 3 provides the Architecture of Revised Proppian's Story Generation System. Section 4 describes semantic reasoning of stories and Section 5 discusses the results and finally section 6 provides the conclusion and future enhancements.

2 Literature Survey

This section details the existing story generators and their strengths and weakness. Propp [2] discussed the story generation as; a tale is a whole that may be composed of thirty one moves. A move is a type of development proceeding from villainy or a lack, through intermediary functions to marriage, or to other functions employed as a denouement (ending). One tale may be composed of several moves that are related between them. One move may directly follow another, but they may also interweave; a development, which has begun pauses and a new move, is inserted. Bailey [3] described an approach to automatic story generation based on the twin assumptions that

it is possible for the generation of a story to be driven by modeling of the responses to the story of an imagined target reader, and that doing so allows the essence of what makes a story work (its ‘storiness’) to be encapsulated in a simple and general way. Charles, F et al [4], presented results from a first version of a fully implemented story-telling prototype, which illustrates the generation of variants of a generic storyline. These variants result from the interaction of autonomous characters with one another, with environment resources or from user intervention.

Dimitrios N. Konstantinou et al [5] discussed about the story generation model HOMER. It receives natural language input in the form of a sentence or an icon corresponding to a scene from a story and it generates a textonly narrative apart from a story line and it includes a plot, characters, settings, the user’s stylistic preferences and also their point-of-view. Riedl et al [6] had provided planning algorithm for story generation. The story planners are limited by the fact that they can only operate on the story world provided, which impacts the ability of the planner to find a solution story plan and the quality and structure of the story plan if one is found, but which lacks semantics.

Jaya et al [7] discussed about the simple story generation by generating suitable sentences with language grammar. The readers or users can select the characters, location, settings; theme for constructing the simple story whereas this model does not concentrates on the semantics of the story. By extending their work [8], the user can conceive the theme by selecting the order of events in new perspective way to construct the stories. In story generation model, the reasoning process begins to reason the sentences in the story to provide the meanings and to check the consistency of the concepts in the ontology. By Integrating the reasoning process for Revised proppians story generation system, helps to improve the quality of the story generated by the System.

3 Revised Proppian’s Story Generation System (RPG)

The Proppian’s model consists of 31 consecutive “*functions*” based on the analysis of Russian folk tales, and they are related to narrate the story with drama roles. Propp’s list of thirty-one generic functions encompassed the entire plot components from which fairy tales were constructed. A cohesive story can be formed by connecting a series of any set of the thirty-one functions in any order. Since each function has upto 10 different possible writings, the computer randomly selects one for each function and outputs them in a sequence and narrates them as a story. The seven drama roles are villain, donor, helper, princess/sought-for-person, dispatcher, hero, and false hero along with the functions. The problem here is that, Propp’s generator fails to recognize the importance of story components such as tone, mood, characterization, writing style etc. A good story should be as specific as possible in every aspect, which includes the attributes of the person playing a role in the story, the time, location and content. Fig. 1 depicts sample story generated by existing Proppians story generation system.



Fig. 1. Existing Proppian Story generator

The working of the Proppian’s generator is described as , each function has an array which stores one possible implementation of the function per cell. When the user hits and submits selected functions, it generates a random number (from 0-9) for each function. It then uses that number to look-up the contents of the corresponding cell in the proper array and stores it in a variable. Finally, it outputs the variables one after another. Hence, Neil [10] discussed that Proppian’s story generator randomly generates a fairy tale from the selected functions. But, it fails to concentrate on situation, their related attributes, and environment. It yields story only by producing set of sentences without having coherent relation. The redundant set of sentences may occur to form the story.

4 Revised Proppians’s Story Generator

The revised Proppians’s system provides an environment to generate a kids’ story with respect to user’s choice. Based on the selection of characters, location, settings, objects, the function will generate a story based on the choice. For example, if the selected setting is ‘forest’, then all the related locations are forest oriented places such as bush, tree top, cave, den, anthills etc; if the selected setting is ‘Zoo’, then the related locations are, cage, tree top, lawn etc are considered for story generation as setting and related locations are considered for character movements. It utilizes canned sequences of the text in order to represent the story. Proppians analyzed Russian folk tales and devised 31 functions, whereas the revised Proppian system describes only five key functions to generate simple stories for kids. The five functions are:

Introduction – This function will provide the initial settings for story generation. Based on the user’s choice, the location and settings are selected and they are represented in the form of a text to give an introduction to the story.

Rise in action – During the normal course of flow or normal settings, characters involved in the story may induce certain actions or interactions with other characters, which may lead to the next step of the story. This function represents the sequence of actions by the characters selected, which helps to lead the flow of the story.

Climax – The climax is the peak of the action. It is nothing but the culmination of the sequence of action: Any sequence of actions comes to the highest or most intense point in the development or resolution of something.

Reaction – The climax comes at the end of any sequence of action. The action which comes after the climax of the story is called reaction. The effect of the reaction depends on the nature of the climax, such as, it may be optimistic or pessimistic; the optimistic climax gives happiness to everyone; the pessimistic climax gives sadness to the reader.

End – It is the final phase of the story which gives the conclusion to any sequence of action. The following algorithm describes the procedure for Story generation.

```

Algorithm Revised Propp (functions, characters,
location, settings, objects)
{// start
Step - 1: retrieve the main characters, settings,
location, objects
Step - 2: Repeat the steps 'a' to 'c' until all the
five functions delivered a text.
  {// function loop
'a' : Repeat the loop (i) to (iii) until it finds the
suitable text.
  { // selecting text from function
    (i) Extract the list of texts related to the
given noun from the array of the list.
    (ii) Extract the texts which match with the
location and settings.
    (iii) Select the suitable text and assign it to
the function variable to form a story.
  } // end of text selection
'b' : Output the function variable as a story
'c' : Move on to the next function if it exists, for
selecting a text. } // end of function loop

```

The 'Extract' function is used for selecting the suitable canned sequence of a text based on the input given.

```

Algorithm Extract (list of canned sequence, noun,
location, setting)
{
Step 1: Tokenize the sequence of the text
Step 2: Match the noun, locations and setting with the
tokens and consider the frequency count of each.
Step 3: Select the texts which have the maximum count
of the noun, location and settings.
Step 4: Suggest the suitable text as output.
} // end of extract function algorithm

```

Kids' stories have been devised from five functions. The user can select the number of characters, attributes of the characters, location etc. The revised Proppian functions

have an array of segments for story generation. The main difference here is, it presents the text based on the description of the user.

Character: wild animal, **settings:** winter season, **location:** Forest. The introduction function may contain the following segment options:

Segment [1] - It was winter day. The climate is cool. Lion was sleeping in the den.

Segment [2] - During the winter day, grasshopper was searching for food.

Segment [3] - In the winter day, donkey was searching for food.

The above are the sample segments available in the 'Introduction' functions. Based on the choice given, wild animal, cold climate, forest location, segment [1] will be selected for the narration of the story. If the user gives the character as domestic animal, then segment [3] will be selected. For example consider the suitable function for the given characters.

Character: grasshopper, ant, **settings:** winter season, **location:** Forest.

The suitable text is selected from the array of segments and given as outputs. Each function depicts the suitable segment for the given inputs. The sample story generated is given below:

Introduction - It was a cold winter day. A grasshopper was very hungry, as the grasshopper does not have anything to eat.

Rise in action - So the grasshopper went to the ant. Grasshopper requested food to ant.

Climax - Ant refused to give the food to grasshopper. Ant gathered food during the entire summer and stored it for the winter.

Reaction - Grasshopper did not do like that.

End - So Grasshopper suffered a lot.

The generated story was, *"It was a cold winter day. A grasshopper was very hungry, and grasshopper does not have anything to eat. So the grasshopper went to the ant. The grasshopper requested food to ant. Ant refused to give the food to grasshopper. Ant gathered food during the entire summer and stored it for the winter. The grasshopper did not do like that."*

The Figure 2 shows the input screen followed by the sample story generated by the revised Proppians's story generator. The revised system produced the sentences based on the choice given by the user. Some of the basic assumptions here are: It Uses the canned sequence of the text for representing a new text. It utilizes the predefined order of events to generate a story. Based on the syntactic match, the relevant text segment is retrieved and there is no conceptual correlation between the characters and events.

Proof Theory:

S.No	Steps of execution	Reason
1.	$\forall x Bird(x) \Rightarrow fly(x)$	Premises
2.	$Bird(c)$	Premises
3.	$Bird(c) \Rightarrow fly(c)$	US, Universal specification
4.	$fly(c)$	2,3 – Rules of detachment

Thus, from the above statement, the reasoner can conclude that “crow can fly”.

Case (ii): To prove “Tiger cannot fly”

Statements: a. All animals cannot fly , b. Some birds can fly. c. Tiger is an animal

Notations: a. $\forall x A(x) \Rightarrow \neg fly(x)$ b. $\exists x B(x) \Rightarrow fly(x)$ c. $A(c)$

Proof Theory:

S.No	Steps of execution	Reason
1.	$\forall x A(x) \Rightarrow \neg fly(x)$	Premises
2.	$A(c)$	Premises
3.	$\forall x A(x) \Rightarrow \neg fly(x)$	US, Universal specification
4.	$\neg fly(x)$	2,3 – Rules of detachment

Thus, from the above statement, the reasoner can conclude that “A Tiger cannot fly “. Fig. 3 shows the stories analyzed by semantic reasoner and the quality of the story discussed in the next section.

6 Results

The quality of the story is determined by the user’s evaluation based on the attributes of the story. Hence, the generated stories need to be analyzed to improve their quality further. Charles James [4] proposed the grading factors for evaluating stories in all the aspects which are listed below in Table 1.

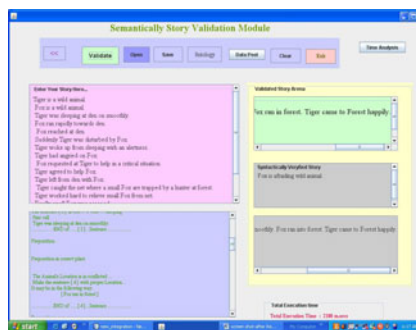


Fig. 3. Stories generated by the RPG – (After Reasoning)

Table 1. Assessment of stories generated by the RPG - (after reasoning)

S.No.	Parameters	Before Reasoning	After Reasoning	% or improvement
1.	Overall – (How is the story as an archetypal fairy tale?)	3.5	3.8	6
2.	Style – (How would you rate the syntactic quality?)	3.3	3.4	2
3.	Grammaticality - How would you rate the syntactic quality	3.7	4.2	10
4.	Flow - Did the sentences flow from one to the next?	3.5	3.9	8
5.	Diction - How appropriate were the author’s word choices?	4.1	4.2	2
6.	Readability - How hard was it to read the prose?	3.5	3.8	6
7.	Logicity - Did the story seem to be out of order	3.6	3.9	6
8.	Believability - Did the story’s characters behave as you would expect them too?	3.8	4.0	4
Average		3.62	3.9	5.5

These factors are evaluated with the following scale such as Excellent – 5; Very Good – 4; Good -3; Fair – 2; needs improvement – 1; and the maximum weightage is 5. The generated story was given to a group of readers and computational linguistics and their rating was consolidated. The generated stories were analyzed in two phases, such as, before and after reasoning. Table 1 gives the user evaluation report for stories generated by RPG and Fig. 4 shows the graphical representation of the quality of story. Since the stories are generated from canned sequence of texts, there is not much change in the syntactic structure. All the factors have been slightly improved, because of semantic checking. It helps to check the conceptual consistency, logicity, believability, overall content and flow of the story. Before reasoning, the RPG system secures 72.4% and after reasoning, this system scores 78%. This RPG model is slightly improved approximately by 5.5% because of semantic reasoning.

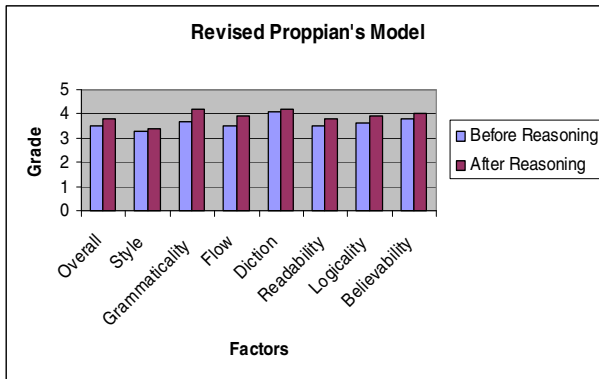


Fig. 4. Assessment of Stories generated by the RPG – (After Reasoning)

7 Conclusion

The ASG system may generate a random sequence of sentences in the story that lacks in semantic flow, consistency of concepts, structure of sentences etc. Hence, conceptual reasoning has been implemented in the story generation system for reasoning the stories using First order logic and Ontology. Ontology provides the semantic description of the story world that contains a detailed description of the characters, objects, location etc. This system enforces the reasoning on the stories by means of consistency checking, subsumption checking and restrictions using ontology. Also, the semantic reasoner checks the syntactic structure of the sentences based on the language grammar. It detects the sentences which do not have suitable nouns, verbs, and prepositions, and suggests the necessary changes to rectify the errors. Semantic reasoning helps to improve the quality of the story. Further, language generation can be incorporated instead of using predefined sentences.

References

1. Bilasco, I.M., Gensel, J., Villanova-Oliver, M.: STAMP: A Model for Generating Adaptable Multimedia Presentations. *Int. J. Multimedia Tools and Applications* 25(3), 361–375 (2005)
2. Propp, V.: *Morphology of the Folktale*. University of Texas Press (1968)
3. Bailey, P.: Searching for storiness: Story generation from a Reader's perspective. In: *Symposium on Narrative Intelligence*. AAAI Press, Menlo Park (1999)
4. Charles, F., Mead, S.J., Cavazza, M.: Character-driven story generation in interactive storytelling. In: *Proceedings of Seventh International Conference on Virtual Systems and Multimedia*, October 25-27, pp. 609–615 (2001)
5. Konstantinou, D.N., Mc Kevitt, P.: HOMER: An Intelligent Multi-modal Story Generation System, Research plan. Faculty of Informatics, University of Ulster, Magee, Londonderry
6. Riedl, M., Young, R.M.: From Linear Story Generation to Branching Story Graphs, pp. 23–29. AAAI, Menlo Park (2005), <http://www.aaai.org>
7. Jaya, A., Sathishkumar, J., Uma, G.V.: A Novel Semantic Validation Mechanism For Automatic Story Generation Using Ontology. In: *The 2007 International Conference on Artificial Intelligence (ICAI 2007)*, Los Vegas, USA, June 25-28 (2007)
8. Jaya, A., Uma, G.V.: An Intelligent System for Automatic Story Generation for Kids Using Ontology. In: *International conference on ACM – Compute 2010*. Organized by ACM Bangalore (accepted for publication)
9. Mark Riedl, O., León, C.: Toward Vignette-Based Story Generation for Drama, Management Systems. In: *2nd International Conference on Intelligent Technologies for Interactive Entertainment, Workshop on Integrating Technologies for Interactive Story* (2008)
10. McIntyre, N., Lapata, M.: Learning to Tell Tales: A Data-driven Approach to Story Generation. In: *Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP*, Singapore, pp. 217–225 (2009)
11. Lipschutz, S.: *Theory and problems of Essential Computer Mathematics*, 1st edn. Schaum's Series (1987/1997)
12. Gruber, T.R.: Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human- Computer Studies* 43(4-5), 907–928 (1995)

A Genetic Algorithm with Entropy Based Probabilistic Initialization and Memory for Automated Rule Mining

Saroj, Kapila, Dinesh Kumar, and Kanika

Department of Computer Science and Engineering
Guru Jambheshwar University of Science and Technology
Hisar-125001, Haryana
ratnoo.saroj@gmail.com, kkishu99@gmail.com,
dinesh_chutani@yahoo.com, kanika.cse@gmail.com

Abstract. In recent years, Genetic Algorithms (GAs) have shown promising results in the domain of data mining. However, unreasonably long running times due to the high computational cost associated with fitness evaluations dissuades the use of GAs for knowledge discovery. In this paper we propose an enhanced genetic algorithm for automated rule mining. The proposed approach supplements the GA with an entropy based probabilistic initialization such that the initial population has more relevant and informative attributes. Further, the GA is augmented with a memory to store fitness scores. The suggested additions have a twofold advantage. Firstly, it lessens the candidate rules' search space making the search more effective to evolve better fit rules in lesser number of generations. Secondly, it reduces number of total fitness evaluations required giving rise to a gain in running time. The enhanced GA has been employed to datasets from UCI machine learning repository and has shown encouraging results.

Keywords: Genetic algorithm, Rule mining, Entropy bases probabilistic initialization, Heap.

1 Introduction

Automated rule mining has attracted a special interest from the data mining research community and several techniques like Decision Trees, Neural Networks, Bayesian Nets, K Nearest Neighbors and Genetic Algorithms have been suggested for discovering classification rules. Genetic Algorithms (GAs) are randomized parallel search algorithms, based on the principal of natural evolution, to search large and not so well defined search spaces. A simple GA starts its search from a randomly initialized population of candidate solutions of the problem under consideration and employs operators like selection, crossover and mutation to explore multidimensional parameter space for an optimal solution [9]. In recent years, GAs have been extensively applied in the domain of Knowledge Discovery and Data Mining. As compared to other rule induction algorithms, GAs have shown promising results in the field of automated rule mining because GAs have the capability to avoid convergence to local optimal solutions and take care of attribute interactions while evolving rules [1, 5, 6,

7, 11, 14, 15 16]. One of the biggest disadvantages of applying GAs for rule mining is that each fitness evaluation is computationally very expensive as it needs to scan the dataset. The tremendous computational demands of fitness evaluation prevent researchers from employing GAs in the field of data mining. Several authors have suggested the use of Parallel Genetic Algorithms to overcome efficiency constraints of sequential GAs. However, implementing PGAs require advanced parallel hardware and programming skills which are not available in many academic institutions and research places.

In this paper we have considered how we can enhance the performance of genetic algorithms for rule mining without using any parallel data mining techniques. Here, we propose a novel and enhanced genetic algorithm approach for automated rule mining which performs better on efficacy as well as on efficiency fronts. The approach employs an entropy based filter to bias the initial population towards more relevant or informative attributes so that the GA starts with better fit rules covering many training instances. A better fit initial population is likely to converge to more reliable rule set with high predictive accuracy in lesser number of generations. In addition, the GA is augmented with a long term memory in the form of heap to store the fitness scores of the rules generated during the evolution. There are more and more duplicate rules generated as GA heads for convergence to an optimal rule set. The fitness for any duplicate rule generated is not re-evaluated but retrieved from the heap. In data mining applications with large datasets, this approach is anticipated to reduce the significant number of fitness evaluations resulting into a considerable gain on efficiency front. The rest of the paper is organized as below.

Section 2 takes a look at the important research attempts to enhance the performance of GAs. Section 3 describes design of the enhanced GA in terms of entropy based probabilistic initialization of the population, evaluation of the fitness of candidate rules and application of genetic operator. The experimental design and results are presented in Section 4. Section 5 concludes the paper and points out the future scope for this work.

2 Related Work

One of the important approaches to improve the reliability and efficiency of genetic algorithms in the domain of Knowledge Discovery is to select relevant attributes as a pre-processing step to discover useful and valid knowledge quicker [2, 3, 4,10]. Selection of relevant attributes reduces the candidate rule space for GA, enhances the efficacy as well as the efficiency of a genetic algorithm and discovers comprehensible rules with higher predictive accuracy [2, 13, 17, 18]. Entropy is a well accepted measure of uncertainty of an object condition. The features with low entropy have a high level of information for prediction. Therefore, entropy measure has been productively employed for feature selection as well as evaluating fitness of rules [2, 3, 4, 5, 17]. We have used the well tested entropy measure to compute the probability of initialization for each attribute.

Random initialization of a population of an evolutionary rule mining algorithm very often generates individual rules that hardly cover any of the training instances. Thus the initial rules in the population have zero or extremely low fitness making the selection process a random walk. In such circumstances, a GA either gives a poor set

of rules or takes unreasonably long running time. To address this problem researchers have suggested a biased initialization process to start with an initial population which is not so far away from the acceptable solutions [1, 5, 10]. One solution to the problem is seeding the initial population i.e. one can choose a training instance to act as a seed for rule generation. This seeded rule can be generalized to cover more number of examples from the dataset [11]. Another approach to ensure some good rules in the initial population is to bias the rule initialization process so that general rule with fewer conditions are more likely to be generated than longer rules [6]. Though general rules in the initial population are expected to cover more training sample, however, in general, there is no guarantee that the randomly generated rules will have enough good rules to lead the search towards an optimal set of rules. Therefore, we have employed an entropy based filtering bias to ensure that the relevant attributes get initialized with higher probability giving rise to an initial population with relatively superior average fitness.

The efficiency of genetic algorithms has also been enhanced by augmenting it with a long term memory so that the solutions/fitness scores evaluated during the evolutionary process are stored in a database and instead of solving a problem from scratch, the previously stored solutions/fitness scores are utilized for later search [8, 12, 19]. In this context, Sushil and Gong (1997) injected the previous solutions stored in a database for similar problems in the initial population to enhance the efficiency of their GA to solve the Traveling Salesman Problem. A memory augmented genetic algorithm has also been presented for optimization of problems that have discrete as well as continuous design variables. This novel GA approach successfully employs a binary search tree as its memory, reduces the number of the fitness evaluations and makes the search more effective, rapidly improving the fitness scores from generation to generation [8].

3 Design of the Enhanced GA

The GA design contains a population initialization module, fitness computation and application of appropriate genetic operators. We have used a GA with crowding to avoid convergence to a single best rule. Rule are discovered as Production Rules in the form of 'If <Premise> Then <Decision>'. Production Rules are high level symbolic rules and are considered comprehensible. The remaining details of GA are as given below.

3.1 Population Initialization

In this work, the initial population is not randomly generated. Instead initial rules contain attributes that are more relevant with higher probability. The stepwise initialization process is as follows.

1. Calculate the entropy of each attribute A_i in the dataset using the following formula.

$$e_{A_i} = - \sum_{j=1}^v \frac{|s_j|}{|s|} * \sum_{k=1}^m p_k \log_2 p_k$$

Where e_{A_i} is the entropy of i^{th} attribute, v is the number of values the attribute A_i can take, m is the number of classes, $l_j / |l|$ acts as the weight of j^{th} partition and the second term gives the expected information required to classify an example from datasets based on partition by attribute A_i .

- Apply a linear transformation on the entropy values of attributes such that an attribute with minimum entropy gets initialized with probability one and an attribute with maximum entropy has a zero probability in the initialization process of the GA population. The figure and the formula to compute the probability of initialization of attribute $A_i(p)$ are given below.

$$A_i(p) = \frac{-1 \times e_{A_i} + e_{A_i}(\max)}{e_{A_i}(\max) - e_{A_i}(\min)}$$

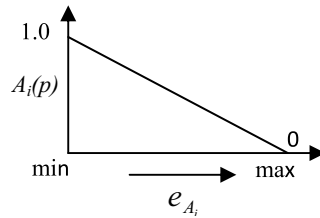


Fig. 1. Computation of probabilities of initialization of attributes

- Generate the initial population probabilistically giving a positive bias to more relevant attributes. We have encoded the chromosome as a fixed length string. A rule is represented in ‘If P Then D’ form; where P is premise/antecedent and is a conjunction of attribute value pairs. Decision D is a single term that contains the value for the goal attribute. All the predicting and goal attribute values are taken as categorical. The locus of each gene is fixed and it gets initialized by one of its alleles. The ‘#’ symbol is used as ‘don’t care’ state. For example consider the description of the well known Weather dataset in Table 1.

Table 1. Description of the Weather dataset

Locus	Attributes	Possible values	Alleles
0	Outlook	Sunny, Cloudy ,Rain	‘s’, ‘c’, ‘r’
1	Temperature	Hot, Mild, Cool	‘h’, ‘m’, ‘c’
2	Humidity	Humid, Normal	‘h’, ‘n’
3	Wind	Windy, Not-windy	‘w’, ‘n’
4	Plan	Volleyball, Swimming, W-lifting	‘1’, ‘2’, ‘3’

Assuming three decision values ‘volleyball’, ‘swimming’ and ‘weight-lifting’ encoded by ‘1’, ‘2’, and ‘3’, the chromosomes for few rules are given below.

Chromosomes/Genotype	Rules/Phenotype
sh##2	If (Outlook=Sunny & Temperature=Hot) Then (decision=Swimming)
r###3	If (Outlook=Rain) Then (decision=Weight-lifting)

3.2 Fitness Computation

The choice of the fitness function is of great importance as it leads the search by evolutionary algorithms towards the optimal solution. Several fitness functions have been suggested for discovery of accurate and interesting rules. These fitness functions take coverage, confidence factor, support, simplicity, entropy and predictive accuracy etc. into consideration [6]. In this work, fitness for a rule r_i for class D_k is computed as per the formula given below.

$$fitness(r_i) = \frac{(confidence(r_i) \times support(r_i))}{simplicity}$$

$$confidence(r_i) = \frac{|P \wedge D_k|}{|P|} \quad support(r_i) = \frac{|P \wedge D_k|}{n}$$

In the above fitness formula, ‘simplicity’ is the number of the attribute value pair in the antecedent part of the rule. The above fitness function favors valid and short rules.

3.3 GA Operators

Roulette wheel is used as the fitness proportionate selection operator. Genetic material of candidate rules can be manipulated/recombined and new genetic material can be introduced using genetic operators like crossover and mutation. We have used a simple one point crossover operator. Mutation operator applied in this work may specialize or generalize a candidate rule by inserting or removing conditional clauses in the antecedent part of the rule. It can also mutate value of an attribute or an attribute itself [6, 7].

3.4 Augmenting GA with Long Term Memory

A heap is an efficient data structure where insertion and deletion of an element take $O(\log_2 n)$ time; where n is number of elements in the heap. We have used heap as a long term memory for the GA to remember all the candidate rules and their associated fitness computations till current generation. A heap as an augmented memory for GA has great practical value as the required data can be stored and retrieved very efficiently. During evolution when a new rule is generated, the heap is searched for the rule. If the rule is found, its fitness is retrieved from heap rather than computing afresh from the dataset which saves a lot of time. If search in the heap fails, then the fitness of the rule is computed and inserted in the heap. This way GA keeps all its fitness computations in its long term memory. A binary tree as a GA memory has been used in [7]. However, a binary tree is not suitable because the fitness values of candidate rules get better and better with increasing number of generations and this is bound to deteriorate the performance of binary tree to a linear linked list. We have used a heap insert to store a new fitness score and a linear search for retrieving a rule’s fitness. The justification for linear search is that a heap has a sequential storage and the average linear search time would be $\ll O(n/2)$.

4 Experimental Design and Results

The proposed GA approach with an entropy based probabilistic initialization and heap as an associative memory is implemented using GALIB247 on a Pentium core 2 duo processor with Ubuntu release 9.10 as operating system. The performance of the suggested approach is validated on three datasets from UCI machine learning repository. The datasets are described in Table 2.

Table 2. Description of datasets used for experimentation

Sr. No	Name of the dataset	No. of Examples	No. of Attributes	No. of Classes
1	Vote	232	16	2
2.	Mushroom	5610	22	2
3.	Nursery	12960	9	4

The crossover and mutation probabilities are kept fixed at 0.66 and 0.1 throughout the datasets. Each data set is unique in terms of number of attributes, alleles and number of examples. The Mushroom and Vote datasets have relatively large candidate rules search space with more number of attributes and alleles as compared to the Nursery data set. The population size and number of maximum generations for each dataset varies and is shown in Table 3.

Table 3. Parameter settings

Sr. No	Name of the dataset	Pop size	No. of gens. (GARI [*])	No. of gens. (GAEBPI [*])
1	Vote	100	100	50
2.	Mushroom	100	2000	200
3.	Nursery	50	100	50

*GARI- GA with Random Initialization.

*GAEBPI- GA with Entropy Based Probabilistic Initialization.

It is not possible to give the rules discovered for all the datasets due to space constraints. However, rules discovered for the Mushroom dataset are given in Table 4. Every dataset was divided into training (50%) and test (50%) datasets and predictive accuracy was computed as given in Table 5. We have also provided predictive accuracy of rule sets discovered by WEKA using ID3 and Multi-layer Perceptron (MP) Classifiers for comparison. We can see that the predictive accuracies are comparable for the Vote and Mushroom datasets but it is less in case of the Nursery dataset. This can be attributed to the fact that our approach discovers much lesser number of rules at a user defined threshold for support and confidence.

With entropy based probabilistic initialization, a GA is able to start with an initial population consisting of better fit rules which is corroborated by Table 6. All the mean and best scores are taken as an average over 20 GA runs.

Table 4. Rules Discovered for Mushroom dataset

Sr. No.	Decision rules Discovered (PRs)	Support	Confidence	Fitness
1.	<i>If</i> (odor = none) <i>Then</i> edible	0.477	0.969	0.462
2.	<i>If</i> (odor = pungent) <i>Then</i> poisonous	0.046	1.0	0.046
3.	<i>If</i> (odor = almond) <i>Then</i> edible	0.071	1.0	0.071
4.	<i>If</i> (odor = foul) <i>Then</i> poisonous	0.282	1.0	0.282
5.	<i>If</i> (odor = creosote) <i>Then</i> poisonous	0.034	1.0	0.034
6.	<i>If</i> (odor = anise) <i>Then</i> edible	0.071	1.0	0.0713
7.	<i>If</i> (odor = musty) <i>Then</i> poisonous	0.0028	1.0	0.0028
8.	<i>If</i> (odor = none && spore_print_color = green) <i>Then</i> poisonous	0.0128	1.0	0.064

Table 5. Predictive accuracy

Sr. No	Name of the dataset	Predictive accuracy (%)	Predictive accuracy (%)	Predictive accuracy (%)
		Our approach	ID3	MP
1	Vote	95.7	93.96	93.1
2.	Mushroom	99.7	100	100
3.	Nursery	89.8	98	99.9

Table 6. Best and mean scores of initial populations

Sr. No	Name of the dataset	Best Score	Best Score	Mean Score	Mean Score
		Random initialization	Biased initialization	Random initialization	Biased initialization
1	Vote	0.065	0.137	0.0021	0.0095
2.	Mushroom	0.0	0.0089	0.0	0.00012
3.	Nursery	0.0185	0.333	0.00045	0.0080

Offline performance of genetic algorithm is taken as the average of best scores whereas online performance gives the average of all scores over the number of generations. As we are interested in an optimal rule set and not in a single best rule, we have considered online performance and the mean score of current population for comparison among the simple GA with Random Initialization (GARI), GA with Entropy based Probabilistic Initialization (GAEBPI), GA with Heap as Augmented Memory (GAHAM) and GA with Entropy based Probabilistic Initialization along with Heap (GAEBPIH). The graphs in Figs 2 and 3 demonstrate that the GA with biased initial population performs better from beginning to end of the evolutionary process as compared to GA with a randomized initial population¹. It is important to note that GAEBPI arrives at better quality of rules in much lesser number of generations.

¹ MSRI- Mean score with Random Initialization; MSEBPI- Mean score with Entropy Based Probabilistic Initialization OLPRI- On-line Performance with Random Initialization; OLPEBPI- On-line Performance Entropy Based Probabilistic Initialization.

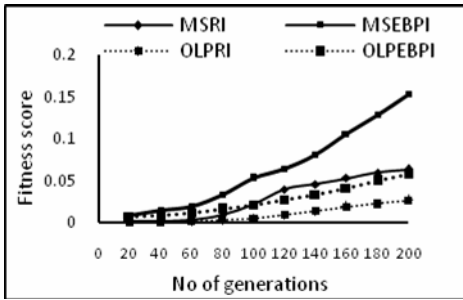


Fig. 2. Performance for Mushroom dataset *No of gen. (GARI)=10*No. of Gens.(GAEBPI)

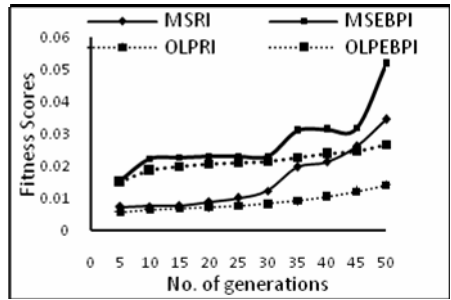


Fig. 3. Performance for Nursery dataset *No of gen. (GARI)=2*No. of Gens.(GAEBPI)

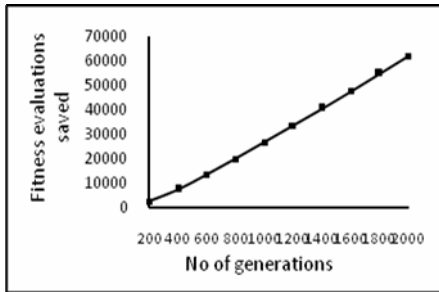


Fig. 4. No. of fitness evaluation saved with GAHAM for Mushroom dataset

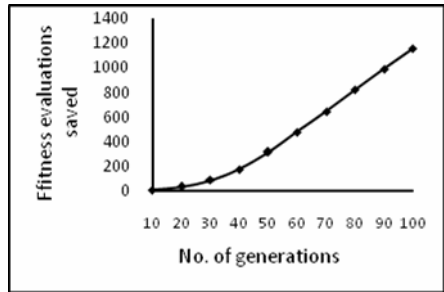


Fig. 5. No. of fitness evaluation saved with GAHAM Nursery dataset

It is clear from the graphs in Figs. 4 and 5 that the number of fitness evaluations saved on account of GAHAM considerably increase with number of generations. The average running time over twenty runs for the three datasets are shown in Table 7. The running time for Multilayer Perceptron (MP) and Multilayer Perceptron with Feature Selection (MPFS) are also provided.

Table 7. Average running time (milliseconds)

Dataset Name	GARI	GAEBPI (gain)	GAHAM (gain)	GAHAM with EBPI (gain)	MP	MPFS
Vote	91	47(1.94)	109(0.83)	44(2.06)	1200	220
Mushroom	55926	4021(13.9)	45906(1.2)	3858(14.5)	5312	26550
Nursery	2257	1240(1.8)	832(2.7)	131(17.2)	146590	17940

An EBPI combined with heap improves the efficiency of the genetic algorithm for all the three datasets. A gain of 13.9 on account of EBPI is noteworthy for Mushroom dataset. This dataset has 22 attributes with more alleles as compared to other two datasets. The huge rule search space is significantly reduced with EBPI. The GAEBPI is able to converge in lesser number of generations giving rise to a substantial speed

gain. The Nursery dataset with 8 predicting attributes takes the least advantage of biased initialization. However, it gets a speed gain of 2.7 with heap employed. The rule search space for this dataset is much smaller as compared to the number of tuples (12960) in it. Therefore, time saved on reduced fitness evaluations dominates the time involved in managing heap. For the Vote dataset the running time increases slightly when a heap is employed. This is due to the fact that Vote dataset contains only 232 examples and 16 attributes each with two alleles. It is a very small dataset as compared to its candidate rule search space. We have more entries in heap and therefore the time for searching the candidate rules in heap dominates the time saved on reduced fitness evaluations. Our approach gives a significant improved time performance as compared to Multilayer Perceptron and Multilayer Perceptron with Feature Selection.

5 Conclusions

An improved genetic algorithm with an entropy based probabilistic initialization along with an associative memory in the form of heap has been proposed for automated rule mining in this paper. The enhanced genetic algorithm is able to discover better rule sets in lesser number of generations and avoids redundant fitness evaluations, thereby, improving the running time. The approach is tested on three datasets of various sizes and has shown promising results. The Entropy Based Probabilistic Initialization is effective with datasets which have large candidate rule search space. Augmenting memory to GA is effective with datasets that have large number of examples so that the fitness evaluation time dominates the memory search overhead. The proposed algorithm is yet to be tested on other datasets to make further generalizations. One of the limitations of the suggested approach is that it works only for discretized attributes and needs to be extended for continuous attributes. It would be interesting to make a comparison between our approach without any parallelism and Parallel Genetic Algorithm approaches that have been implemented for automated rule mining on clusters of workstations using message passing techniques like PVM or MPI.

References

1. Alatas, B., Akin, E.: An Efficient Genetic Algorithm for Automated Mining of both Positive and Negative Quantitative Association Rules. *Soft Computing – A Fusion of Foundations, Methodologies and Application* 10, 230–237 (2005)
2. Balamurugan, A., Pramala, R.: Rajaram: Feature Selection for Large Scale Data by Combining Class Association Rule Mining and Information Gain: a Hybrid Approach. *Inter-networking Indonesia Journal* 1, 17–20 (2009)
3. Dash, M., Liu, H.: Feature Selection for Classification. *Intelligent Data Analysis: An International Journal* 3, 131–156 (1997)
4. Dash, M., Choi, K., Scheuermann, P., Liu, H.: Feature Selection for Clustering-A Filter Solution. In: *Proceedings of IEEE Second International Conference on Data Mining*, pp. 115–122. IEEE Press, New York (2002)

5. Dhar, V., Chou, D., Provost, F.: Discovering Interesting Patterns for Investment Decision Making with GLOWER- A Genetic Learner Overlaid With Entropy Reduction. *Data Mining and Knowledge Discovery* 4, 251–280 (2000)
6. Freitas, A.A.: *Data Mining and Knowledge Discovery with Evolutionary Algorithm*. Natural Computing Series. Springer, New York (2002)
7. Freitas, A.A.: A survey of Evolutionary Algorithms for Data Mining and Knowledge Discovery. In: *Advances in Evolutionary Computation Theory and Applications*, pp. 819–845. Springer, New York (2003)
8. Gantovnik, V.B., Anderson-Cook, C.M., Gurdal, Z., Watson, L.T.: A Genetic Algorithm with Memory for Mixed Discrete–Continuous Design Optimization. *Computers and Structures* 81, 2003–2009 (2003)
9. Goldberg, D.E.: *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Company Inc., New York (1989)
10. Kohavi, R., John, G.H.: Wrappers for Feature Subset Selection. *Artificial Intelligence* 97, 273–324 (1997)
11. Liu, J.J., Kwok, J.T.Y.: An Extended Genetic Rule Induction Algorithm. In: *Proceeding of the Congress on Evolutionary Computation (CEC 2000)*, pp. 458–463. IEEE Press, La Jolla (2000)
12. Louis, S.J., Li, G.: Augmenting Genetic Algorithms with Memory to Solve Salesman Problems. In: *Proceedings of the Joint Conference on Information Sciences*, pp. 108–111. Duke University Press (1997)
13. Martín-Bautista, M.J., Vila, M.-A.: Applying Genetic Algorithms to the Feature Selection Problem in Information Retrieval. In: *Andreasen, T., Christiansen, H., Larsen, H.L. (eds.) FQAS 1998. LNCS (LNAI), vol. 1495*, pp. 272–281. Springer, Heidelberg (1998)
14. Noda, E., Freitas, A.A., Lopes, H.S.: Discovering Interesting Prediction Rules with a Genetic Algorithm. In: *Proc. Congress on Evolutionary Computation (CEC)*, Washington, pp. 1322–1329 (1999)
15. Saroj, B., K. K.: A Parallel Genetic Algorithm Approach for Automated Discovery of Censored Production Rules. In: *Proc. IASTED Int. Conf. on Artificial Intelligence and Application*, pp. 435–441. ACTA Press, Innsbruck (2007)
16. Saroj, B., K.K.: Discovery of Exceptions: A Step towards Perfection. In: *Third International Conference on Network and System Security*, Gold Coast, Australia, pp. 540–545 (2009)
17. Subbotin, S., Oleynik, A.: Entropy Based Evolutionary Search for Feature Selection. In: *9th International Conference the Experience of Designing and Application of CAD System in Microelectronic*, Polyana, Ukraine, pp. 442–443 (2007)
18. Tan, F., Fu, X., Zhang, Y., Bourgeois, A.G.: A Genetic Algorithm-Based Method for Feature Subset Selection. *Soft Computing* 12, 111–120 (2008)
19. Yang, S.: Associative memory scheme for genetic algorithms in dynamic environments. In: *Rothlauf, F., Branke, J., Cagnoni, S., Costa, E., Cotta, C., Drechsler, R., Lutton, E., Machado, P., Moore, J.H., Romero, J., Smith, G.D., Squillero, G., Takagi, H. (eds.) EvoWorkshops 2006. LNCS, vol. 3907*, pp. 788–799. Springer, Heidelberg (2006)

Author Index

- Abd El-Haleem, Ahmed M. I-236
Abraham, Jibi II-383
Achary, K.K. I-265
Aeron, Anurag II-126
Agarwal, Ajay I-189
Agarwal, Harit I-491
Agarwal, Vishal II-107
Agrawal, P.K. III-280
Agrawal, Vishnu P. I-398
Ahlawat, Savita III-56
Ajith, B. II-55
Akerkar, Rajendra III-35
Alagarsamy, K. II-321
Ali, Ihab A. I-236
Amin, Mohamed I-147
Ananthanarayana, V.S. III-85
Ananthapadmanabha, T. III-400
Anitha, R. III-237
Appavu, Subramanian I-501
Arif, Mohammad II-223, II-464
Arora, Sparsh II-564
Arthi, R. II-148
Arunmozhi, S.A. III-210
Arya, K.V. I-1
Ashish, Tanwer II-617
Ashok Baburaj, C. II-321
Atishay, Jain II-617

Bag, Soumen I-358
Balachandra I-158
Balakrishanan, G. I-43
Bali, Rasmeet S. II-179
Baneerjee, P.K. I-217
Bapat, Jyotsna II-633
Baras, John S. II-88
Barbhuiya, Ferdous A. II-432, II-472
Baskaran, K. II-349
Basu, Saikat II-491
Bathia, Pranjal III-268
Beerelli, Bharath Reddy III-268
Ben Ghezala, Henda I-439
Bermúdez, Aurelio III-325
Bhadkoliya, Pratik I-452
Bhadra Chaudhuri, S.R. II-422
Bhat, Ganesh I-265
Bhat, Narasimha B. I-49
Bhatia, Divya II-564
Bhatt, Mohnish II-383
Bhattacharjee, Anup K. I-125, III-139
Bhattacharya, Swapan I-70
Bhattacharya, Tanmay II-422
Bhattacharyya, D.K. III-76
Bhoopathybagan, K. I-307, III-290
Bhuvanewswaran, R.S. I-278, II-70
Bhuyan, Monowar H. III-76
Bikash, Sharma II-372
Biradar, Rajashekhar C. II-33
Biswas, S. II-432

Casado, Rafael III-325
Chaki, Rituparna I-33
Chakrabarti, Indrajit III-108
Chakraborty, Kaushik II-215
Chakraborty, Sandip II-472
Chand, Phool I-125, III-139
Chanda, Jayeeta I-70
Chandersekar, Coimbatore III-217
Chandorkar, Nirbhay I-125
Chandrakumar, Jeyamala II-516
Chandra Sekaran, K. I-49
Chaniara, B.P. II-107
Chattopadhyay, Santanu I-90, I-168, III-410
Chaudhary, Ankit III-46
Chaudhary, Sachin II-223
Chaurasia, Alok I-452
Chavan, Hariram I-523
Chavan, V.T. I-248
Chawda, Dushyant III-173
Chellappan, C. II-169
Chen, Tsung Teng I-428
Chi, Yen Ping I-428
Chiu, Yaw Han I-428
Choudhary, Amit III-56

Dahiya, Ratna III-438
Damodaram, A. II-290
Dananjayan, P. II-535
Dandash, Osama II-410
Das, Apurba I-532

- Das, Karen III-46
 Das, Rama Krushna III-161
 Dasgupta, K.S. II-107
 Datta, Raja II-400
 David, Jisa II-391
 Debasish, Bhaskar II-372
 Deepa, R. II-349
 Deepthy, G.S. II-1
 Despande, Bharat M. I-398
 Devaraj, P. I-278, II-70
 Devarakonda, Nagaraju I-101
 Dey, Haimabati II-400
 Dilo, Arta II-595
 Divyanshu, II-202
 D'Mello, Demian Antony III-85
 Dodda, Sandhyarani III-310
 Doreswamy, I-512
 Durga, Toshniwal I-24, I-551,
 II-444, III-358
 Dutta, Shreyash K. I-388
 Dwivedi, Sudhanshu III-450
- El-Sawy, Abdel Rahman H. I-236
 Elwahsh, Haitham I-147
- García, Eva M. III-325
 Gayathri, S. II-584
 Goel, Aayush II-55
 Goel, Anshul III-450
 Goel, Mohit II-546
 Gohil, Gunvantsinh I-348
 Gohokar, V.V. II-137
 Gopalakrishnan, K. I-135
 Govardhan, A. I-101
 Goyal, Ruchita II-202
 Goyani, Mahesh M. I-339, I-348, I-388
 Grace, L.K. Joshila III-459
 Guimarães, Almir P. II-302
 Gupta, B.B. III-280
 Gupta, Indranil Sen I-168
 Gupta, Manjari I-318
 Gupta, Siddharth III-258
- Hari Narayanan, R. II-573
 Harit, Gaurav I-358
 Hashem, Mohamed I-147
 Hatai, Indranil III-108
 Havinga, Paul J.M. II-595
 Hemanth, K.S. I-512
- Hemnani, Khushboo III-173
 Hency, Berlin II-233
 Hore, Sirshendu II-422
 Howlader, Jaydeep II-491
 Hubballi, N. II-432
- Ibrahim, Ibrahim I. I-236
 Ithnin, Norafida I-209
- Jagadeesh, B.S. I-125, III-139
 Jain, Ankit II-546
 Jana, Prasanta K. I-329
 Janakiraman, T.N. II-329, II-645
 Jaya, A. I-594
 Jena, Jayadev II-313
 Jeyabalan, Jeyalakshmi II-20
 Jeyakumar, G. I-472
 Joseph, M. I-49
 Joshi, Brijesh I-339
 Joshi, Gauri III-450
 Joshi, R.C. III-280
 Joshi, Shantanu II-243
- Kakkasageri, M.S. II-254
 Kalita, J.K. III-76
 Kalmady, Rajesh I-125, III-139
 Kamal, Ankit III-421
 Kanika, I-604
 Kanisha, Johnny I-43
 Kanjilal, Ananya I-70
 Kapania, Ashish II-362
 Kapila, I-604
 Karimi, Ramin I-209
 Karnani, Urvashi III-139
 Kaur, Ravinder II-117
 Keshavamurthy, B.N. I-24, II-444,
 III-358
 Khader, Sheik Abdul III-190
 Khaja Muhaiyadeen, A. II-573
 Khan, Ibrahim III-200
 Kiran, H.S. Shashi I-378
 Koli, S.M. II-137
 Kosta, Yogeshwar P. III-338
 Kraiem, Naoufel I-439
 Krishna, C. Rama II-126, II-179
 Krishnan, Avinash III-378
 Krishnan, S. Murali II-63
 Kshirsagar, S.P. II-137
 Kulkarni, Prakash Jayanth I-462
 Kumar, Binod II-55

- Kumar, Dinesh I-256, I-604
 Kumar, G. Charan II-44
 Kumar, Manish II-55
 Kumar, M. Chenthil III-183
 Kumar, Mondal Arun I-217
 Kumar, Mukesh II-627
 Kumar, Santosh II-453
 Kumar, Sarkar Subir II-372
 Kumar, Sumit II-453
 Kumar, Vijay I-256
 Kumari, V. Valli I-481
 Kumar Sahu, Prasanna II-313
 Kundu, Santanu I-90
 Kushwaha, D.S. I-491
- Lakshmi, Rajya I-59
 Lal, Mohan II-126
 Laverdière, Marc-André III-268
 Le, Kim III-300
 Le, Phu Dung II-410
 Loganathan, Priya II-20
- MacGregor, Mike H. II-656
 Maciel, Paulo R.M. II-302
 Madan, D.K. II-627
 Madhavan, Kavitha I-577
 Mahesh, P.K. I-368
 Maheswari, V. III-459
 Majumder, Soumyadip II-472
 Malik, Jyoti III-438
 Manjula, R. III-237
 Manjula, V. II-169
 Manjunath, Aishwarya III-378
 Mankad, Kunjal III-35
 Manna, Kanchan I-168
 Manvi, Sunilkumar S. II-33, II-254
 Masoum, Alireza II-595
 Mathur, Alok I-491
 Matias Jr., Rivalino II-302
 Mazumdar, Debasis I-532
 Meghanathan, Natarajan I-14, II-606
 Mehra, Rajesh II-117
 Mehta, S.V. II-107
 Meratnia, Nirvana II-595
 Mishra, Abhipsa II-99
 Mishra, Arun I-452
 Mishra, Deepak II-191
 Mishra, Devender I-329
 Mishra, Manoj II-202
- Mishra, Shakti I-491
 Mishra, Shivani II-502
 Mishra, Subhankar II-99
 Misra, Arun K. I-452, I-491
 Misra, Manoj III-280
 Misro, Ajita Kumar III-161
 Mitra, Soma I-532
 Mukherjee, Ayan II-422
 Mukherjee, Nilarun I-560
 Mulherkar, Jaideep III-450
 Murugan, K. II-148, II-158, II-482
- Nagabhushan, B.S. II-556
 Nagamalai, Dhinaharan III-459
 Nagammai, M. I-501
 Nagaraju, Aitha II-44
 Nagaraju, S. I-512
 Nagasimha, M P II-383
 Nageshkumar, M. I-298
 Naik, Chaitanya II-383
 Nair, Vivek II-491
 Najafzadeh, Sara I-209
 Namala, Murali Babu I-378
 Nandagopal, Malarvizhi III-149
 Nandi, Sukumar II-243, II-432, II-453, II-472
 Narayanamurthy, Gopalakrishnan III-1
 Narayanamurthy, Vigneswaran III-1
 Nath, Sur Samarendra II-372
 Nedunchezian, Raghavendran II-233
 Ngo, Huy Hoang II-410
- Ouali, Sami I-439
- Pakala, Hara Gopal Mani III-200
 Palanisamy, P. I-179
 Palsule, V.S. II-107
 Pamidi, Srinivasulu I-101
 Pande, Akshara I-318
 Pandey, Bipul III-66
 Pardeshi, Bharat I-551
 Pareek, Narendra K. I-413
 Parthibarajan, Aravinthan III-1
 Parthibarajan, Arun Srinivas III-1
 Parua, Suparna I-532
 Parveen, Katheerja III-190
 Parvez, Moin I-59
 Pateriya, Pushpendra Kumar II-502
 Patidar, Vinod I-413

- Patil, A.R. I-248
 Patil, Kiran Kumari II-556
 Patnaik, Sachidananda III-161
 Patnaik, Sumagna III-120
 Piramuthu, Selwyn II-357, III-431
 Pradhan, Sambhu Nath III-410
 Prasad, B.D.C.N. I-570
 Prasad, P.E.S.N. Krishna I-570
 Prema, K.V. I-158
 Priyanga, N. I-501
 Priyanka, S. I-501
 Purandare, R.G. II-137
 Purohit, G.N. II-10, III-367
- Rabbani, Munir Ahamed III-190
 Rabindranath, Bera II-372
 Rahamatkar, Surendra I-189
 Raheja, J.L. III-46
 Raheja, Sonia III-46
 Raj, Payal I-388
 Rajanikanth, K. II-265
 Rajaram, Ramasamy I-225, I-501
 Rajesh, G. II-573
 Rajeswari, S. I-112
 Raju, K.V.S.V.N. I-481, III-200
 Ram, Anant I-542
 Ramachandram, S. II-44
 Ramachandran, Selvakumar III-130,
 III-310
 Ramachandran, V. II-432
 Ramaswamy, T.V. I-90
 Rangaswamy, T.M. II-55
 Ranjan, Prabhat III-450
 Rao, Ajay I-158
 Rao, Rajwant Singh I-318
 Rao, Santhosha I-80
 Rao, V.V.R. Maheswara I-481
 Rasappan, Suresh I-585
 Ratti, R. II-432
 Ravish Aradhya, H.V. II-362
 Rayudu, Haritha III-130
 Raziuddin, Syed I-59
 Reddy, Damodar I-329
 Reddy, Geetha J. III-378
 Reddy, K. Yashwant II-613
 Rishi, Rahul II-627, III-56
 Rishiwal, Vinay I-1
 Robert Masillamani, M. II-340
 Roopa, S. II-432
- Sabyasachi, Samanta II-523
 Sagar, Yeruva I-570
 Saha, Himadri Nath II-215
 Saha, Soumyabrata I-33
 Sainarayanan, G. III-438
 Sajja, Priti Srinivas III-35
 Saleena, B. III-183
 Salian, Supriya III-85
 Sam, I. Shatheesh I-278, II-70
 Samaddar, Shefalika Ghosh II-502
 Samuel, K. Deepak II-63
 Sandhya, M.K. II-482
 Sane, Suneeta I-523
 Santapoor, Lavanya III-130, III-310
 Sarangapani, Usha II-20
 Sarkar, Mohanchur II-107
 Sarma, Abhijit II-243
 Saroj, I-604
 Sasirekha, G.V.K. II-633
 Sathishkumar, G.A. I-307, III-290
 Sathya Priya, S. II-158
 Satija, Kavita II-223
 Satpathy, Sudhansu Mohan II-99
 Sattar, Syed Abdul I-59
 Saurabh, Dutta II-523
 Schlegel, Christian II-656
 Selvanayagam, Raman II-516
 Selvi, R. Muthu I-225
 Sen, Jaydip III-247
 Sengupta, Abhrajit II-215
 Sengupta, Sabnam I-70
 Senthil Thilak, A. II-329, II-645
 Shabana, II-464
 Shah, Hemal III-338
 Shama, Kumara I-80
 Shankar, Shobha III-400
 Shanmugavelayutham, C. I-472
 ShanmukhaSwamy, M.N. I-298, I-368
 Sharada, A. III-390
 Sharma, Deepak II-546
 Sharma, Krishna Gopal I-542
 Sharma, Usha III-367
 Sharvani, G.S. II-55
 Shelton Paul Infant, C. II-573
 Shetty, Shravan I-378
 Shikkenawis, Gitam I-339
 Shobha, K.R. II-265
 Shukla, Anupam III-66
 Shyjila, P.A. I-288

- Simpson, William R. III-217
 Singh, Ajit II-627
 Singh, Pratibha III-10
 Singh, Sanjay I-378
 Singh, Sunil Kumar III-258
 Singh, Yashpal I-542
 Sirbi, Kotrappa I-462
 Somasundaram, Kiran K. II-88
 Soumyasree, Bera II-372
 Sreenivasan, Rajagopal II-633
 Srinivasan, Bala II-410
 Sriraam, N. I-307, III-290
 Srivatsa, S.K. III-183
 Subbusundaram, Balasundaram II-584
 Subramanyan, B. II-516
 Sud, Krishan K. I-413
 Suganthi, K. II-63
 Sunitha, K.V.N. III-390
 Sur, A. II-432
 Susan, R.J. II-1
 Swati, Chowdhuri I-217
 Swetha, P.M. I-179

 Tabassum, Kahkashan II-290
 Taghikhaki, Zahra II-595
 Talekar, Sopan A. III-470
 Tandon, Puneet II-191
 Taruna, S. II-10
 Thangakumar, J. II-340
 Thomas, Antu Annam II-277, III-23
 Thomas, Ciza II-391
 Tidake, Vaishali S. III-470
 Tiwari, Ritu III-66

 Tomar, Minal III-10
 Tripathi, A.K. I-318

 Uma, G.V. I-594
 Upadhyay, Nitin I-398
 Uthariaraj, V. Rhymend I-135, I-199,
 II-79, III-149, III-348

 Vaidyanathan, Sundarapandian I-577,
 I-585, III-98
 Valli, R. II-535
 Valli Kumari, V. I-101
 Vanathi, B. I-199, II-79, III-348
 Vasal, Apurv II-191
 Vasantha Kumar, N.G. II-107
 Vashishtha, Rohit I-491
 Vastrad, Channabasayya M. I-512
 Venkatachalam, SP. III-290
 Venkataramani, Y. I-112, III-210
 Venkatesh, Vasudha II-233
 Verma, A.K. II-546
 Verma, Bhupendra III-173
 Verma, Seema III-367
 Vignesh, R. III-290
 Vijay Kumar, B.P. II-556
 Vijaykumar, M. I-80

 Wilscy, M. I-288, II-277, III-23
 Wilson, Campbell II-410

 Yadav, Mano I-1

 Zadeh, Parisa D. Hossein II-656