

Bijaya Ketan Panigrahi  
Ponnuthurai Nagarathnam Suganthan  
Swagatam Das  
Suresh Chandra Satapathy (Eds.)

LNCS 7077

# Swarm, Evolutionary, and Memetic Computing

Second International Conference, SEMCCO 2011  
Visakhapatnam, Andhra Pradesh, India, December 2011  
Proceedings, Part II

2  
Part II

 Springer

*Commenced Publication in 1973*

Founding and Former Series Editors:

Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

David Hutchison

*Lancaster University, UK*

Takeo Kanade

*Carnegie Mellon University, Pittsburgh, PA, USA*

Josef Kittler

*University of Surrey, Guildford, UK*

Jon M. Kleinberg

*Cornell University, Ithaca, NY, USA*

Alfred Kobsa

*University of California, Irvine, CA, USA*

Friedemann Mattern

*ETH Zurich, Switzerland*

John C. Mitchell

*Stanford University, CA, USA*

Moni Naor

*Weizmann Institute of Science, Rehovot, Israel*

Oscar Nierstrasz

*University of Bern, Switzerland*

C. Pandu Rangan

*Indian Institute of Technology, Madras, India*

Bernhard Steffen

*TU Dortmund University, Germany*

Madhu Sudan

*Microsoft Research, Cambridge, MA, USA*

Demetri Terzopoulos

*University of California, Los Angeles, CA, USA*

Doug Tygar

*University of California, Berkeley, CA, USA*

Gerhard Weikum

*Max Planck Institute for Informatics, Saarbruecken, Germany*

Bijaya Ketan Panigrahi  
Ponnuthurai Nagaratnam Suganthan  
Swagatam Das Suresh Chandra Satapathy (Eds.)

# Swarm, Evolutionary, and Memetic Computing

Second International Conference, SEMCCO 2011  
Visakhapatnam, Andhra Pradesh, India, December 19-21, 2011  
Proceedings, Part II

Volume Editors

Bijaya Ketan Panigrahi  
IIT Delhi, New Delhi, India  
E-mail: bkpanigrahi@ee.iitd.ac.in

Ponnuthurai Nagarathnam Suganthan  
Nanyang Technological University, Singapore  
E-mail: epnsugan@ntu.edu.sg

Swagatam Das  
Jadavpur University, Kolkata, India  
E-mail: swagatamdas19@yahoo.co.in

Suresh Chandra Satapathy  
ANITS, Visakhapatnam, India  
E-mail: sureshsatapathy@gmail.com

ISSN 0302-9743 e-ISSN 1611-3349  
ISBN 978-3-642-27241-7 e-ISBN 978-3-642-27242-4  
DOI 10.1007/978-3-642-27242-4  
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2011943108

CR Subject Classification (1998): F.1, I.2, J.3, F.2, I.5, I.4

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

© Springer-Verlag Berlin Heidelberg 2011

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.

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

*Typesetting:* Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

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

# Preface

This LNCS volume contains the papers presented at the Second Swarm, Evolutionary and Memetic Computing Conference (SEMCCO-2011) held during December 19–21, 2011 at Anil Neerukonda Institute of Technology and Sciences (ANITS), Visakhapatnam, Andhra Pradesh, India. SEMCCO is regarded as one of the prestigious international conference series that aims at bringing together researchers from academia and industry to report and review the latest progress in cutting-edge research on swarm, evolutionary, memetic computing and other novel computing techniques, to explore new application areas, to design new bio-inspired algorithms for solving specific hard optimization problems, and finally to create awareness of these domains to a wider audience of practitioners.

SEMCCO-2011 received 422 paper submissions in total from 25 countries across the globe. After a rigorous peer-review process involving 1,025 reviews in total, 124 full-length articles were accepted for oral presentation at the conference. This corresponds to an acceptance rate of 25% and is intended for maintaining the high standards of the conference proceedings. The papers included in this LNCS volume cover a wide range of topics in swarm, evolutionary, memetic and other intelligent computing algorithms and their real-world applications in problems selected from diverse domains of science and engineering.

The conference featured four distinguished keynote speakers. Carlos. A. Coello Coello's talk on "Recent Results and Open Problems in Evolutionary Multi-objective Optimization" reviewed some of the research topics on evolutionary multi-objective optimization that are currently attracting a lot of interest (e.g., many-objective optimization, hybridization, indicator-based selection, use of surrogates, etc.) and which represent good opportunities for doing research. Jacek M. Zurada in his talk on "prediction of Secondary Structure of Proteins Using Computational Intelligence and Machine Learning Approaches with Emphasis on Rule Extraction" emphasized the Prediction of protein secondary structures (PSS) with discovery of prediction rules underlying the prediction itself. He explored the use of C4.5 decision trees to extract relevant rules from PSS predictions modeled with two-stage support vector machines (TS-SVM). Dipankar Dasgupta delivered his keynote address on "Advances in Immunological Computation." N.R. Pal's talk on "Fuzzy Rule-Based Systems for Dimensionality Reduction" focused on the novelty of fuzzy rule-based systems used for dimensionality reduction through feature extraction preserving the "original structure" present in high-dimensional data.

SEMCCO-2011 also included two invited talks and tutorial, which were free to all conference participants. The invited talks were delivered by Sumanth Yenduri, University of Southern Mississippi, and Amit Kumar, CEO and Chief Scientific Officer, Bio-Axis DNA Research Center, Hyderabad, on the topics "Wireless Sensor Networks—Sink Shift Algorithms to Maximize Efficiency" and "Eval-

uating Mixed DNA Evidence with Forensic Bioinformatics,” respectively. The tutorial was delivered by Siba K. Udgata of the University of Hyderabad, India, on “Swarm Intelligence: An Optimization Tool for Various Engineering Applications.” The tutorial gave a brief overview of many swarm intelligence algorithms. The talk also covered an in-depth comparative study of these algorithms in different domains. In particular, emphasis was given to engineering applications like clustering in data mining, routing in networks, node placement in wireless sensor networks, finding the shortest path for packet forwarding, optimum resource allocation and planning, software failure prediction in software engineering, among many others.

We take this opportunity to thank the authors of all submitted papers for their hard work, adherence to the deadlines and patience with the review process. The quality of a refereed volume depends mainly on the expertise and dedication of the reviewers. We are thankful to the reviewers for their timely effort and help rendered to make this conference successful. We are indebted to the Program Committee members who not only produced excellent reviews but also constantly encouraged us during the short time frames to make the conference of international repute.

We would also like to thank our sponsors for providing all the support and financial assistance. First, we are indebted to ANITS Management and Administrations (The Secretary and Correspondent, the Principal and Directors and faculty colleagues and administrative personnel of the Departments of CSE, IT and MCA) for supporting our cause and encouraging us to organize the conference at ANITS, Vishakhapatnam. In particular, we would like to express our heart-felt thanks to Sri V. Thapovardhan, Secretary and Correspondent, ANITS, for providing us with the necessary financial support and infrastructural assistance to hold the conference. Our sincere thanks are due to V.S.R.K. Prasad, Principal, ANITS, for his continuous support. We thank Kalyanmoy Deb, IIT Kanpur, India, and Lakhmi Jain, Australia, for providing valuable guidelines and inspiration to overcome various difficulties in the process of organizing this conference as General Chairs. We extend our heart-felt thanks to Janusz Kacprzyk, Poland, for guiding us as the Honorary Chair of the conference. The financial assistance from ANITS and the others in meeting a major portion of the expenses is highly appreciated. We would also like to thank the participants of this conference, who have considered the conference above all hardships. Finally, we would like to thank all the volunteers whose tireless efforts in meeting the deadlines and arranging every detail ensured that the conference ran smoothly.

December 2011

Bijaya Ketan Panigrahi  
Swagatam Das  
P.N. Suganthan  
Suresh Chandra Satapathy

# Organization

## Chief Patron

Sri V.Thapovardhan Secretary and Correspondent , ANITS

## Patrons

V.S.R.K. Prasad Principal, ANITS  
Govardhan Rao Director (Admin), ANITS  
K.V.S.V.N. Raju Director (R & D), ANITS

## Organizing Chairs

S.C. Satapathy HoD of CSE, ANITS  
Ch. Suresh HoD of IT, ANITS  
Ch. Sita Kameswari HoD of MCA, ANITS

## Honorary Chair

Janusz Kacprzyk Poland

## General Chairs

Kalyanmoy Deb IIT Kanpur , India  
Lakhmi Jain Australia

## Program Chairs

B.K. Panigrahi Indian Institute of Technology (IIT), Delhi,  
India  
Swagatam Das Jadavpur University, Kolkata, India  
Suresh Chandra Satapathy ANITS, India

## Steering Committee Chair

P.N. Suganthan Singapore

## Publicity / Special Session Chair

Sanjoy Das, USA

Zhijia Cui, China

Wei-Chiang Samuelson Hong, Taiwan

## International Advisory Committee

Almoataz Youssef Abdelaziz, Egypt

Athanasios V. Vasilakos, Greece

Boyang Qu, China

Carlos A. Coello Coello, Mexico

Chilukuri K. Mohan, USA

Delin Luo, China

Dipankar Dasgupta, USA

Fatih M. Tasgetiren, Turkey

Ferrante Neri, Finland

G.K. Venayagamoorthy, USA

Gerardo Beni, USA

Hai Bin Duan, China

Heitor Silvério Lopes, Brazil

J.V.R. Murthy, India

Jane J. Liang, China

Janez Brest, Slovenia

Jeng-Shyang Pan, Taiwan

Juan Luis Fernández Martínez, USA

K. Parsopoulos, Greece

Kay Chen Tan, Singapore

Leandro Dos Santos Coelho, Brazil

Ling Wang, China

Lingfeng Wang, China

M.A. Abido, Saudi Arabia

M.K. Tiwari, India

Maurice Clerc, France

Namrata Khemka, USA

Oscar Castillo, Mexico

Pei-Chann Chang, Taiwan

Peng Shi, UK

P.V.G.D. Prasad Reddy, India

Qingfu Zhang, UK

Quanke Pan, China

Rafael Stubs Parpinelli, Brazil

Rammohan Mallipeddi, Singapore

Roderich Gross, UK

Ruhul Sarker, Australia

S. Baskar, India

S.K. Udgata, India

S.S. Dash, India

S.S. Pattanaik, India

S.G. Ponnambalam, Malaysia

Saeid Nahavandi, Australia

Saman Halgamuge, Australia

Shizheng Zhao, Singapore

X.Z. Gao, Finland

Yew Soon Ong, Singapore

Ying Tan, China

Zong Wo Alex K. Qin, France

Amit Konar, India

Amit Kumar, India

Anupam Shukla, India

Ashish Anand, India

Damodaram A., India

D.K. Chaturvedi, India

Dilip Pratihari, India

Dipti Srinivasan, Singapore

Frank Neumann, Australia

G.S.N. Raju, India

Hong Yan, Hong Kong

Jeng-Shyang Pan, Taiwan

John MacIntyre, UK

Ke Tang, China

M. Shashi, India

Meng Joo Er., Singapore

Meng-Hiot Lim, Singapore

Oscar Castillo, Mexico

P.K. Singh, India

P.S. Avadhani, India

Rafael Stubs Parpinelli, Brazil

Richa Sing, India

Robert Kozma, USA

R. Selvarani, India

Sachidananda Dehuri, India



Samuelson W. Hong, Taiwan  
 Sumanth Yenduri, USA  
 Suresh Sundaram, Singapore  
 V. Kamakshi Prasad, India

V. Sree Hari Rao, India  
 Yucheng Dong, China

## Technical Review Board

Clerc Maurice  
 M. Willjuice Iruthayarajan  
 Janez Brest  
 Zhihua Cui  
 Millie Pant  
 Sidhartha Panda  
 Ravipudi Rao  
 Matthieu Weber  
 Q.K. Pan  
 Subramanian Baskar  
 V. Ravikumar Pandi  
 Krishnand K.R.  
 Jie Wang  
 V. Mukherjee  
 S.P. Ghoshal  
 Boyang Qu  
 Tianshi Chen  
 Roderich Gross  
 Sanyou Zeng  
 Ashish Ranjan Hota  
 Yi Mei  
 M. Rammohan  
 Sambrata Dasg  
 S. Miruna Joe Amali  
 Kai Qin  
 Bijan Mishra  
 S. Dehury  
 Shizheng Zhao  
 Chilukuri Mohan  
 Nurhadi Siswanto  
 Aimin Zhou  
 Nitin Anand Shrivastava  
 Dipankar Maity  
 Ales Zamuda  
 Minlong Lin  
 Ben Niu  
 D.K. Chaturvedi  
 Peter Korošec

Mahmoud Abdallah  
 Nidul Sinha  
 Soumyadip Roy  
 Anyong Qing  
 Sanyou Zeng  
 Siddharth pal  
 Ke Tang  
 Sheldon Hui  
 Noha Hamza  
 Kumar Gunjan  
 Anna Kononova  
 Noha Hamza  
 Iztok Fister  
 Fatih Tasgetiren  
 Eman Samir Hasan  
 Tianshi Chen  
 Ferrante Neri  
 Jie Wang  
 Deepak Sharma  
 Matthieu Weber  
 Sayan Maity  
 Abdelmonaem Fouad Abdallah  
 Sheldon Hui  
 Kenneth Price  
 Nurhadi Siswanto  
 S.N. Omark  
 Minlong Lin  
 Shih-Hsin Chen  
 Sasitharan Balasubramaniam  
 Aniruddha Basak  
 Shih-Hsin Chen  
 Fatih Tasgetiren  
 Soumyadip Roy  
 S. Sivananathapermal  
 Borko Boskovic  
 Pugalenth Ganesan  
 Ville Tirronen  
 Jane Liang

Ville Tirronen  
Bing Xue  
Andrea Caponio  
S. Sivananaithapermal  
Yi Mei  
Paramasivam Venkatesh  
Saber Elsayed

Saurav Ghosh  
Hamim Zafar  
Saber Elsayed  
Anyong Qing  
Arpan Mukhopadhyay  
Ye Xu

## Organizing Committee

P. Srinivasu  
B. Tirimula Rao  
M. James Stephen  
S. Ratan Kumar  
S. Jayaprada  
B. Ravi Kiran  
K. Neelima Santhoshi  
Ch. Demudu Naidu  
K.S. Deepthi  
Y.V. Srinivasa Murthy  
G. Jagadish  
G.V. Gayathri  
A. Kavitha  
A. Deepthi  
T. Kranthi  
S. Ranjan Mishra  
S.A. Bhavani  
K. Mrunalini  
S. Haleema  
M. Kranthi Kiran

K. Chandra Sekhar  
K. Sri Vaishnavi  
N. Sashi Prabha  
K. Santhi  
G. Gowri Pushpa  
K.S. Sailaja  
D. Devi Kalyani  
G. Santhoshi  
G.V.S. Lakshmi  
V. Srinivasa Raju  
Ch. Rajesh  
N. Sharada  
M. Nanili Tuveera  
Usha Chaitanya  
I. Sri Lalita Sarwani  
K. Yogeswara Rao  
T. Susan Salomi  
P. Lavanya Kumari  
K. Monni Sushma Deep  
S.V.S.S. Lakshmi

## Table of Contents – Part II

Register Allocation via Graph Coloring Using an Evolutionary Algorithm . . . . .	1
<i>Sevin Shamizi and Shahriar Lotfi</i>	
A Survey on Swarm and Evolutionary Algorithms for Web Mining Applications . . . . .	9
<i>Ashok Kumar Panda, S.N. Dehuri, M.R. Patra, and Anirban Mitra</i>	
Exploration Strategies for Learning in Multi-agent Foraging . . . . .	17
<i>Yogeswaran Mohan and S.G. Ponnambalam</i>	
Nurse Rostering Using Modified Harmony Search Algorithm . . . . .	27
<i>Mohammed A. Awadallah, Ahamad Tajudin Khader, Mohammed Azmi Al-Betar, and Asaju La'aro Bolaji</i>	
A Swarm Intelligence Based Algorithm for QoS Multicast Routing Problem . . . . .	38
<i>Manoj Kumar Patel, Manas Ranjan Kabat, and Chita Ranjan Tripathy</i>	
Test Data Generation: A Hybrid Approach Using Cuckoo and Tabu Search . . . . .	46
<i>Krish Perumal, Jagan Mohan Ungati, Gaurav Kumar, Nitish Jain, Raj Gaurav, and Praveen Ranjan Srivastava</i>	
Selection of GO-Based Semantic Similarity Measures through AMDE for Predicting Protein-Protein Interactions . . . . .	55
<i>Anirban Mukhopadhyay, Moumita De, and Ujjwal Maulik</i>	
Towards Cost-Effective Bio-inspired Optimization: A Prospective Study on the GPU Architecture . . . . .	63
<i>Paula Prata, Paulo Fazendeiro, and Pedro Sequeira</i>	
Cricket Team Selection Using Evolutionary Multi-objective Optimization . . . . .	71
<i>Faez Ahmed, Abhilash Jindal, and Kalyanmoy Deb</i>	
Data Clustering Using Harmony Search Algorithm . . . . .	79
<i>Osama Moh'd Alia, Mohammed Azmi Al-Betar, Rajeswari Mandava, and Ahamad Tajudin Khader</i>	

Application of Swarm Intelligence to a Two-Fold Optimization Scheme for Trajectory Planning of a Robot Arm . . . . .	89
<i>Tathagata Chakraborti, Abhronil Sengupta, Amit Konar, and Ramadoss Janarthanan</i>	
Two Hybrid Meta-heuristic Approaches for Minimum Dominating Set Problem . . . . .	97
<i>Anupama Potluri and Alok Singh</i>	
Automatic Clustering Based on Invasive Weed Optimization Algorithm . . . . .	105
<i>Aritra Chowdhury, Sandip Bose, and Swagatam Das</i>	
Classification of Anemia Using Data Mining Techniques . . . . .	113
<i>Shilpa A. Sanap, Meghana Nagori, and Vivek Kshirsagar</i>	
Taboo Evolutionary Programming Approach to Optimal Transfer from Earth to Mars . . . . .	122
<i>M. Mutyalarao, A. Sabarinath, and M. Xavier James Raj</i>	
Solving Redundancy Optimization Problem with a New Stochastic Algorithm . . . . .	132
<i>Chun-Xia Yang and Zhi-Hua Cui</i>	
Energy Efficient Cluster Formation in Wireless Sensor Networks Using Cuckoo Search . . . . .	140
<i>Manian Dhivya, Murugesan Sundarambal, and J. Oswald Vincent</i>	
Data Clustering Based on Teaching-Learning-Based Optimization . . . . .	148
<i>Suresh Chandra Satapathy and Anima Naik</i>	
Extracting Semantically Similar Frequent Patterns Using Ontologies . . . . .	157
<i>S. Vasavi, S. Jayaprada, and V. Srinivasa Rao</i>	
Correlating Binding Site Residues of the Protein and Ligand Features to Its Functionality . . . . .	166
<i>B. Ravindra Reddy, T. Sobha Rani, S. Durga Bhavani, Raju S. Bapi, and G. Narahari Sastry</i>	
Non-linear Grayscale Image Enhancement Based on Firefly Algorithm . . . . .	174
<i>Tahereh Hassanzadeh, Hakimeh Vojodi, and Fariborz Mahmoudi</i>	
Synthesis and Design of Thinned Planar Concentric Circular Antenna Array - A Multi-objective Approach . . . . .	182
<i>Sk. Minhazul Islam, Saurav Ghosh, Subhrajit Roy, Shizheng Zhao, Ponnuthurai Nagaratnam Suganthan, and Swagamtam Das</i>	

Soft Computing Based Optimum Parameter Design of PID Controller in Rotor Speed Control of Wind Turbines . . . . .	191
<i>R. Manikandan and Nilanjan Saha</i>	
Curve Fitting Using Coevolutionary Genetic Algorithms . . . . .	201
<i>Nejat A. Afshar, Mohsen Soryani, and Adel T. Rahmani</i>	
A Parallel Hybridization of Clonal Selection with Shuffled Frog Leaping Algorithm for Solving Global Optimization Problems (P-AISFLA) . . . . .	211
<i>Suresh Chittineni, A.N.S. Pradeep, G. Dinesh, Suresh Chandra Satapathy, and P.V.G.D. Prasad Reddy</i>	
Non-uniform Circular-Shaped Antenna Array Design and Synthesis - A Multi-Objective Approach . . . . .	223
<i>Saurav Ghosh, Subhrajit Roy, Sk. Minhazul Islam, Shizheng Zhao, Ponnuthurai Nagaratnam Suganthan, and Swagatam Das</i>	
Supervised Machine Learning Approach for Bio-molecular Event Extraction . . . . .	231
<i>Asif Ekbal, Amit Majumder, Mohammad Hasanuzzaman, and Sriparna Saha</i>	
Design of Two Channel Quadrature Mirror Filter Bank: A Multi-Objective Approach . . . . .	239
<i>Subhrajit Roy, Sk. Minhazul Islam, Saurav Ghosh, Shizheng Zhao, Ponnuthurai Nagaratnam Suganthan, and Swagatam Das</i>	
Soft Computing Approach for Location Management Problem in Wireless Mobile Environment . . . . .	248
<i>Moumita Patra and Siba K. Udgata</i>	
Distribution Systems Reconfiguration Using the Hyper-Cube Ant Colony Optimization Algorithm . . . . .	257
<i>A.Y. Abdelaziz, Reham A. Osama, S.M. El-Khodary, and Bijaya Ketan Panigrahi</i>	
Bacterial Foraging Optimization Algorithm Trained ANN Based Differential Protection Scheme for Power Transformers . . . . .	267
<i>M. Geethanjali, V. Kannan, and A.V.R. Anjana</i>	
Reduced Order Modeling of Linear MIMO Systems Using Soft Computing Techniques . . . . .	278
<i>Umme Salma and K. Vaisakh</i>	
Statistical and Fusion Based Hybrid Approach for Fault Signal Classification in Electromechanical System . . . . .	287
<i>Tribeni Prasad Banerjee and Swagatam Das</i>	

Steganalysis for Calibrated and Lower Embedded Uncalibrated Images .....	294
<i>Deepa D. Shankar, T. Gireeshkumar, and Hiran V. Nath</i>	
An Efficient Feature Extraction Method for Handwritten Character Recognition .....	302
<i>Manju Rani and Yogesh Kumar Meena</i>	
Optimized Neuro PI Based Speed Control of Sensorless Induction Motor .....	310
<i>R. Arulmozhiyal, C. Deepa, and Kaliyaperumal Baskaran</i>	
Wavelet Based Fuzzy Inference System for Simultaneous Identification and Quantitation of Volatile Organic Compounds Using SAW Sensor Transients .....	319
<i>Prashant Singh and R.D.S. Yadava</i>	
<b>Author Index</b> .....	329

# Table of Contents – Part I

Design of Two-Channel Quadrature Mirror Filter Banks Using Differential Evolution with Global and Local Neighborhoods.....	1
<i>Pradipta Ghosh, Hamim Zafar, Joydeep Banerjee, and Swagatam Das</i>	
Differential Evolution with Modified Mutation Strategy for Solving Global Optimization Problems.....	11
<i>Pravesh Kumar, Millie Pant, and V.P. Singh</i>	
Self-adaptive Cluster-Based Differential Evolution with an External Archive for Dynamic Optimization Problems.....	19
<i>Udit Halder, Dipankar Maity, Preetam Dasgupta, and Swagatam Das</i>	
An Informative Differential Evolution with Self Adaptive Re-clustering Technique.....	27
<i>Dipankar Maity, Udit Halder, and Preetam Dasgupta</i>	
Differential Evolution for Optimizing the Hybrid Filter Combination in Image Edge Enhancement.....	35
<i>Tirimula Rao Benala, Satchidananda Dehuri, G.S. Surya Vamsi Sirisetti, and Aditya Pagadala</i>	
Scheduling Flexible Assembly Lines Using Differential Evolution.....	43
<i>Lui Wen Han Vincent and S.G. Ponnambalam</i>	
A Differential Evolution Based Approach for Multilevel Image Segmentation Using Minimum Cross Entropy Thresholding.....	51
<i>Soham Sarkar, Gyana Ranjan Patra, and Swagatam Das</i>	
Tuning of Power System Stabilizer Employing Differential Evolution Optimization Algorithm.....	59
<i>Subhransu Sekhar Tripathi and Sidhartha Panda</i>	
Logistic Map Adaptive Differential Evolution for Optimal Capacitor Placement and Sizing.....	68
<i>Kamal K. Mandal, Bidishna Bhattacharya, Bhimsen Tudu, and Niladri Chakraborty</i>	
Application of an Improved Generalized Differential Evolution Algorithm to Multi-objective Optimization Problems.....	77
<i>Subramanian Ramesh, Subramanian Kannan, and Subramanian Baskar</i>	

Enhanced Discrete Differential Evolution to Determine Optimal Coordination of Directional Overcurrent Relays in a Power System . . . . .	85
<i>Joy mala Moirangthem, Subranshu Sekhar Dash, K.R. Krishnanand, and Bijaya Ketan Panigrahi</i>	
Dynamic Thinning of Antenna Array Using Differential Evolution Algorithm . . . . .	94
<i>Ratul Majumdar, Aveek Kumar Das, and Swagatam Das</i>	
A Quantized Invasive Weed Optimization Based Antenna Array Synthesis with Digital Phase Control . . . . .	102
<i>Ratul Majumdar, Ankur Ghosh, Souvik Raha, Koushik Laha, and Swagatam Das</i>	
Optimal Power Flow for Indian 75 Bus System Using Differential Evolution . . . . .	110
<i>Aveek Kumar Das, Ratul Majumdar, Bijaya Ketan Panigrahi, and S. Surender Reddy</i>	
A Modified Differential Evolution Algorithm Applied to Challenging Benchmark Problems of Dynamic Optimization . . . . .	119
<i>Ankush Mandal, Aveek Kumar Das, and Prithwijit Mukherjee</i>	
PSO Based Memetic Algorithm for Unimodal and Multimodal Function Optimization . . . . .	127
<i>Swapna Devi, Devidas G. Jadhav, and Shyam S. Pattnaik</i>	
Comparison of PSO Tuned Feedback Linearisation Controller (FBLC) and PI Controller for UPFC to Enhance Transient Stability . . . . .	135
<i>M. Jagadeesh Kumar, Subranshu Sekhar Dash, M. Arun Bhaskar, C. Subramani, and S. Vivek</i>	
A Nelder-Mead PSO Based Approach to Optimal Capacitor Placement in Radial Distribution System . . . . .	143
<i>Pradeep Kumar and Asheesh K. Singh</i>	
Comparative Performance Study of Genetic Algorithm and Particle Swarm Optimization Applied on Off-grid Renewable Hybrid Energy System . . . . .	151
<i>Bhimsen Tudu, Sibsankar Majumder, Kamal K. Mandal, and Niladri Chakraborty</i>	
An Efficient Algorithm for Multi-focus Image Fusion Using PSO-ICA . . . . .	159
<i>Sanjay Agrawal, Rutuparna Panda, and Lingaraj Dora</i>	
Economic Emission OPF Using Hybrid GA-Particle Swarm Optimization . . . . .	167
<i>J. Preetha Roselyn, D. Devaraj, and Subranshu Sekhar Dash</i>	



Application of Improved PSO Technique for Short Term Hydrothermal Generation Scheduling of Power System . . . . .	176
<i>S. Padmini, C. Christober Asir Rajan, and Pallavi Murthy</i>	
Multi-objective Workflow Grid Scheduling Based on Discrete Particle Swarm Optimization . . . . .	183
<i>Ritu Garg and Awadhesh Kumar Singh</i>	
Solution of Economic Load Dispatch Problem Using Lbest-Particle Swarm Optimization with Dynamically Varying Sub-swarms . . . . .	191
<i>Hamim Zafar, Arkabandhu Chowdhury, and Bijaya Ketan Panigrahi</i>	
Modified Local Neighborhood Based Niching Particle Swarm Optimization for Multimodal Function Optimization . . . . .	199
<i>Pradipta Ghosh, Hamim Zafar, and Ankush Mandal</i>	
Constrained Function Optimization Using PSO with Polynomial Mutation . . . . .	209
<i>Tapas Si, Nanda Dulal Jana, and Jaya Sil</i>	
Rank Based Hybrid Multimodal Fusion Using PSO . . . . .	217
<i>Amiyo Kumar, Madasu Hanmandlu, Vaibhav Sharma, and H.M. Gupta</i>	
Grouping Genetic Algorithm for Data Clustering . . . . .	225
<i>Santhosh Peddi and Alok Singh</i>	
Genetic Algorithm for Optimizing Neural Network Based Software Cost Estimation . . . . .	233
<i>Tirimula Rao Benala, Satchidananda Dehuri, Suresh Chandra Satapathy, and Ch. Sudha Raghavi</i>	
IAMGA: Intimate-Based Assortative Mating Genetic Algorithm . . . . .	240
<i>Fatemeh Ramezani and Shahriar Lotfi</i>	
SVR with Chaotic Genetic Algorithm in Taiwanese 3G Phone Demand Forecasting . . . . .	248
<i>Li-Yueh Chen, Wei-Chiang Hong, and Bijaya Ketan Panigrahi</i>	
Genetic Algorithm Assisted Enhancement in Pattern Recognition Efficiency of Radial Basis Neural Network . . . . .	257
<i>Prabha Verma and R.D.S. Yadava</i>	
An Approach Based on Grid-Value for Selection of Parents in Multi-objective Genetic Algorithm . . . . .	265
<i>Rahila Patel, M.M. Raghuwanshi, and L.G. Malik</i>	
A Novel Non-dominated Sorting Algorithm . . . . .	274
<i>Gaurav Verma, Arun Kumar, and Krishna K. Mishra</i>	

Intelligent Genetic Algorithm for Generation Scheduling under Deregulated Environment . . . . .	282
<i>Sundararajan Dhanalakshmi, Subramanian Kannan, Subramanian Baskar, and Krishnan Mahadevan</i>	
Impact of Double Operators on the Performance of a Genetic Algorithm for Solving the Traveling Salesman Problem . . . . .	290
<i>Goran Martinovic and Drazen Bajer</i>	
Parent to Mean-Centric Self-Adaptation in SBX Operator for Real-Parameter Optimization . . . . .	299
<i>Himanshu Jain and Kalyanmoy Deb</i>	
Attribute Reduction in Decision-Theoretic Rough Set Models Using Genetic Algorithm . . . . .	307
<i>Srilatha Chebrolu and Sriram G. Sanjeevi</i>	
A Study of Decision Tree Induction for Data Stream Mining Using Boosting Genetic Programming Classifier . . . . .	315
<i>Dirisala J. Nagendra Kumar, J.V.R. Murthy, Suresh Chandra Satapathy, and S.V.V.S.R. Kumar Pullela</i>	
Bi-criteria Optimization in Integrated Layout Design of Cellular Manufacturing Systems Using a Genetic Algorithm . . . . .	323
<i>I. Jerin Leno, S. Saravana Sankar, M. Victor Raj, and S.G. Ponnambalam</i>	
Reconfigurable Composition of Web Services Using Belief Revision through Genetic Algorithm . . . . .	332
<i>Deivamani Mallayya and Baskaran Ramachandran</i>	
Neural Network Based Model for Fault Diagnosis of Pneumatic Valve with Dimensionality Reduction . . . . .	341
<i>P. Subbaraj and B. Kannapiran</i>	
A CAD System for Breast Cancer Diagnosis Using Modified Genetic Algorithm Optimized Artificial Neural Network . . . . .	349
<i>J. Dheeba and S. Tamil Selvi</i>	
Application of ANN Based Pattern Recognition Technique for the Protection of 3-Phase Power Transformer . . . . .	358
<i>Harish Balaga, D.N. Vishwakarma, and Amrita Sinha</i>	
Modified Radial Basis Function Network for Brain Tumor Classification . . . . .	366
<i>S.N. Deepa and B. Aruna Devi</i>	
Attribute Clustering and Dimensionality Reduction Based on In/Out Degree of Attributes in Dependency Graph . . . . .	372
<i>Asit Kumar Das, Jaya Sil, and Santanu Phadikar</i>	

MCDM Based Project Selection by F-AHP & VIKOR .....	381
<i>Tuli Bakshi, Arindam Sinharay, Bijan Sarkar, and Subir kumar Sanyal</i>	
Nonlinear Time Series Modeling and Prediction Using Local Variable Weights RBF Network .....	389
<i>Garba Inoussa and Usman Babawuro</i>	
Detection of Disease Using Block-Based Unsupervised Natural Plant Leaf Color Image Segmentation .....	399
<i>Shitala Prasad, Piyush Kumar, and Anuj Jain</i>	
Measuring the Weight of Egg with Image Processing and ANFIS Model .....	407
<i>Payam Javadikia, Mohammad Hadi Dehrouyeh, Leila Naderloo, Hekmat Rabbani, and Ali Nejat Lorestani</i>	
Palmprint Authentication Using Pattern Classification Techniques .....	417
<i>Amioy Kumar, Mayank Bhargava, Rohan Gupta, and Bijaya Ketan Panigrahi</i>	
A Supervised Approach for Gene Mention Detection .....	425
<i>Sriparna Saha, Asif Ekbal, and Sanchita Saha</i>	
Incorporating Fuzzy Trust in Collaborative Filtering Based Recommender Systems .....	433
<i>Vibhor Kant and Kamal K. Bharadwaj</i>	
A Function Based Fuzzy Controller for VSC-HVDC System to Enhance Transient Stability of AC/DC Power System .....	441
<i>Niranjan Nayak, Sangram Kesari Routray, and Pravat Kumar Rout</i>	
A Bayesian Network Riverine Model Study .....	452
<i>Steven Spansel, Louise Perkins, Sumanth Yenduri, and David Holt</i>	
Application of General Type-2 Fuzzy Set in Emotion Recognition from Facial Expression .....	460
<i>Anisha Halder, Rajshree Mandal, Aruna Chakraborty, Amit Konar, and Ramadoss Janarthanan</i>	
Design of a Control System for Hydraulic Cylinders of a Sluice Gate Using a Fuzzy Sliding Algorithm .....	469
<i>Wu-Yin Hui and Byung-Jae Choi</i>	
Rough Sets for Selection of Functionally Diverse Genes from Microarray Data .....	477
<i>Sushmita Paul and Pradipta Maji</i>	

Quality Evaluation Measures of Pixel - Level Image Fusion Using Fuzzy Logic .....	485
<i>Srinivasa Rao Dammavalam, Seetha Maddala, and M.H.M. Krishna Prasad</i>	
Load Frequency Control: A Polar Fuzzy Approach .....	494
<i>Rahul Umrao, D.K. Chaturvedi, and O.P. Malik</i>	
An Efficient Algorithm to Computing Max-Min Post-inverse Fuzzy Relation for Abductive Reasoning .....	505
<i>Sumantra Chakraborty, Amit Konar, and Ramadoss Janarthanan</i>	
Fuzzy-Controlled Energy-Efficient Weight-Based Two Hop Clustering for Multicast Communication in Mobile Ad Hoc Networks.....	520
<i>Anuradha Banerjee, Paramartha Dutta, and Subhankar Ghosh</i>	
Automatic Extractive Text Summarization Based on Fuzzy Logic: A Sentence Oriented Approach .....	530
<i>M. Esther Hannah, T.V. Geetha, and Saswati Mukherjee</i>	
An Improved CART Decision Tree for Datasets with Irrelevant Feature .....	539
<i>Ali Mirza Mahmood, Mohammad Imran, Naganjaneyulu Satuluri, Mrithyumjaya Rao Kuppa, and Vemulakonda Rajesh</i>	
Fuzzy Rough Set Approach Based Classifier .....	550
<i>Alpna Singh, Aruna Tiwari, and Sujata Naegi</i>	
Proposing a CNN Based Architecture of Mid-level Vision for Feeding the WHERE and WHAT Pathways in the Brain .....	559
<i>Apurba Das, Anirban Roy, and Kuntal Ghosh</i>	
Multithreaded Memetic Algorithm for VLSI Placement Problem .....	569
<i>Subbaraj Potti and Sivakumar Pothiraj</i>	
Bacterial Foraging Approach to Economic Load Dispatch Problem with Non Convex Cost Function .....	577
<i>B. Padmanabhan, R.S. Sivakumar, J. Jasper, and T. Aruldoss Albert Victoire</i>	
Static/Dynamic Environmental Economic Dispatch Employing Chaotic Micro Bacterial Foraging Algorithm .....	585
<i>Nicole Pandit, Anshul Tripathi, Shashikala Tapaswi, and Manjaree Pandit</i>	
Artificial Bee Colony Algorithm with Self Adaptive Colony Size.....	593
<i>Tarun Kumar Sharma, Millie Pant, and V.P. Singh</i>	

Multi-Robot Box-Pushing Using Non-dominated Sorting Bee Colony Optimization Algorithm .....	601
<i>Pratyusha Rakshit, Arup Kumar Sadhu, Preetha Bhattacharjee, Amit Konar, and Ramadoss Janarthanan</i>	
Emotion Recognition from the Lip-Contour of a Subject Using Artificial Bee Colony Optimization Algorithm .....	610
<i>Anisha Halder, Pratyusha Rakshit, Aruna Chakraborty, Amit Konar, and Ramadoss Janarthanan</i>	
Software Coverage : A Testing Approach through Ant Colony Optimization .....	618
<i>Bhuvnesh Sharma, Isha Girdhar, Monika Taneja, Pooja Basia, Sangeetha Vadla, and Praveen Ranjan Srivastava</i>	
Short Term Load Forecasting Using Fuzzy Inference and Ant Colony Optimization .....	626
<i>Amit Jain, Pramod Kumar Singh, and Kumar Anurag Singh</i>	
The Use of Strategies of Normalized Correlation in the Ant-Based Clustering Algorithm .....	637
<i>Arkadiusz Lewicki, Krzysztof Pancierz, and Ryszard Tadeusiewicz</i>	
Ant Based Clustering of Time Series Discrete Data – A Rough Set Approach .....	645
<i>Krzysztof Pancierz, Arkadiusz Lewicki, and Ryszard Tadeusiewicz</i>	
Sensor Deployment for Probabilistic Target $k$ -Coverage Using Artificial Bee Colony Algorithm .....	654
<i>S. Mini, Siba K. Udgata, and Samrat L. Sabat</i>	
Extended Trail Reinforcement Strategies for Ant Colony Optimization .....	662
<i>Nikola Ivkovic, Mirko Malekovic, and Marin Golub</i>	
Fractional-Order $PI^{\lambda}D^{\mu}$ Controller Design Using a Modified Artificial Bee Colony Algorithm .....	670
<i>Anguluri Rajasekhar, Vedurupaka Chaitanya, and Swagatam Das</i>	
Reconfiguration of Distribution Systems for Loss Reduction Using the Harmony Search Algorithm .....	679
<i>A.Y. Abdelaziz, Reham A. Osama, S.M. El-Khodary, and Bijaya Ketan Panigrahi</i>	
An Improved Multi-objective Algorithm Based on Decomposition with Fuzzy Dominance for Deployment of Wireless Sensor Networks .....	688
<i>Soumyadip Sengupta, Md. Nasir, Arnab Kumar Mondal, and Swagatam Das</i>	

Application of Multi-Objective Teaching-Learning-Based Algorithm to an Economic Load Dispatch Problem with Incommensurable Objectives .....	697
<i>K.R. Krishnanand, Bijaya Ketan Panigrahi, P.K. Rout, and Ankita Mohapatra</i>	
Application of NSGA – II to Power System Topology Based Multiple Contingency Scrutiny for Risk Analysis .....	706
<i>Nalluri Madhusudana Rao, Diptendu Sinha Roy, and Dusmanta K. Mohanta</i>	
Multi Resolution Genetic Programming Approach for Stream Flow Forecasting .....	714
<i>Rathinasamy Maheswaran and Rakesh Khosa</i>	
Reference Set Metrics for Multi-Objective Algorithms .....	723
<i>Chilukuri K. Mohan and Kishan G. Mehrotra</i>	
Groundwater Level Forecasting Using SVM-QPSO .....	731
<i>Ch. Sudheer, Nitin Anand Shrivastava, Bijaya Ketan Panigrahi, and M Shashi Mathur</i>	
Genetic Algorithm Based Optimal Design of Hydraulic Structures with Uncertainty Characterization .....	742
<i>Raj Mohan Singh</i>	
<b>Author Index</b> .....	751

# Register Allocation via Graph Coloring Using an Evolutionary Algorithm

Sevin Shamizi<sup>1</sup> and Shahriar Lotfi<sup>2</sup>

<sup>1</sup>Department Of Computer, Shabestar Branch, Islamic Azad University, Shabestar, Iran  
Sevin.Shamizi@gmail.com

<sup>2</sup>Computer Science Department, University of Tabriz, Tabriz, Iran  
Shahriar\_Lotfi@tabrizu.ac.ir

**Abstract.** Register allocation is one of most important compiler optimization techniques. The most famous method for register allocation is graph coloring and the solution presented in this paper (RAGCES) is based on the graph coloring too; for coloring interference graph, evolutionary algorithm is used. In this method graph coloring and selecting variables for spilling is taken place at the same time. This method was tested on several graphs and results were compared with the results of the previous methods.

**keywords:** Compiler, Code Optimization, Register Allocation, Graph Coloring and Evolutionary Algorithms.

## 1 Introduction

The results of computations of a program are stored either in memory or in registers. Compared to memory, registers are faster, but they are scarce resources.

When a register is needed for a computation but all available registers are in use, the contents of one of the used registers must be spilled into memory in order to free up a register and then reloaded from memory into a register again when needed and it needs cost. In order to decrease the additional costs those variables must be selected to spill in which they will be using less in the future. Register allocation determines what values in a program should reside in registers. Register allocation is a NP-hard problem.

Register allocation is done in optimization phase of compilers. The optimization goal is to hold as many values as possible in registers in order to avoid expensive memory accesses [1], [8]. In fact register allocation is considered as the problem of mapping symbolic register to a fixed number of physical register and the goal is to find an assignment that minimizes the cost of spill. Many register allocators consider graph coloring as a paradigm to model the register allocation problem. To model the allocation problem, a compiler constructs an interference graph,  $G(V,E)$ , where  $V$  is the set of individual live ranges and  $E$  is the set of edges that represent interferences between live ranges. Fig. 1 is a sample interference graph. The problem of determining whether a graph is K-colorable is NP-complete in general [8], [9], [10].

This paper is going to explain the below:

In the second section is explaining some basic introductions and descriptions of terms used in this paper. Section three is discussing the traditional approaches for register allocation and section four presents some details of suggested method in this paper.

## 2 Terminology

This section explains terms used in this used in this paper [1], [2].

**Live Track:** The live track of variable is the set of consecutive program point that the variable is live.

**Assignment Rule:** two variables live at the same program point cannot be assigned to the same register at that point.

**Conflict:** two tracks have a conflict if they break the assignment rule.

**Spill Cost:** The cost of spilling the variable is depending on the number of definitions and uses. The spill cost of a variable is the estimated runtime cost of the corresponding variable for loading from and storing in memory.

**Evolutionary Algorithms:** Evolutionary Algorithms involve natural evolution and genetics. The genetic algorithm is a classical method in this category, it has been applied to various optimization problems and it is suitable for solving NP-hard problems. There are several other methods like genetic programming, Evolutionary programming and Evolutionary strategies.

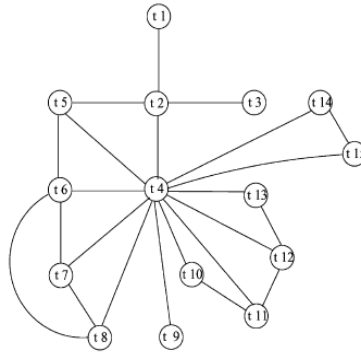
**Graph Coloring:** A graph is considered colored with  $n$  color if all of nodes neighbors (those connected to it by arcs) can be assigned a color different than the nodes color.

## 3 Related Work

This section of the paper is introducing the traditional methods for register allocation in compilers. The most well known method for register allocation is graph coloring [2]. Register allocation by graph coloring first introduced by Chaitin in 1981 then Briggs introduced an optimistic allocation (OCH) that improved coloring strategy by producing better allocation. The main difference between Briggs's algorithm and chaitin's is the timing of the spill decisions. Briggs's allocator makes spill decision later than process done in chaitin's [8], [7].

The chaitin-Briggs algorithm consists of six phases [6], [7], [3], [4]. First phase builds the interference graph. Second phase is coalesce in which register-to-register copies are removed when the source and destination register do not interfere. Third phase calculate spill costs and the simplify phase consults the spill cost and orders the nodes by placing them on a stack. In fifth phase the allocator tries to color the graph by repeatedly popping a node from the stack, inserting in to the graph and attempting to assign it a color. In the last phase spill code is inserted for nodes marked for spilling [3]. Another strategy for register allocation problem is the priority-based method described by Hennessy and chow [5].





**Fig. 1.** A sample Interference graph [9]

Topcuoglu, Demiroz and Kandemir introduced the HEA (Hybrid Evolutionary Algorithm) in 2007. They propose a steady-state GA that takes two input string and their problem-specific crossover operator (GFPX) is applied to generate a single offspring and it is followed by local search operator [9].

## 4 The Proposed Approach (RAGCEA)

The solution used in this paper is based on graph coloring and for graph coloring, Evolutionary Algorithm is used. In this approach coloring and spilling variables are taking place at the same time. This method tries to color the interference graph using  $k$  colors (the number of registers) but when a graph is not colorable with  $k$  color, the algorithm spills one or more variables. Algorithm tries to spill the least number of nodes and also it tries to spill those which have low spill costs. In order to implement this algorithm we added one more color (zero). Each node which colored with zero will be spilled. The details of proposed algorithm are presented in this section.

### 4.1 Coding

In order to present a chromosome, array is used. A chromosome show a colored graph and spilled nodes at the same time. For this case an array with the length of number of graph nodes is used. In each columns of array the color of nodes is given and the nodes with zero color will be spilled. In each columns of array the color of nodes is given and the nodes with zero color will be spilled. Fig. 2 shows a sample chromosome for graph in fig. 1.

1	2	3	...	15										
1	0	3	2	1	4	1	1	2	3	0	1	2	1	4

**Fig. 2.** Sample chromosome for graph given in fig.1

## 4.2 Fitness Function

The fitness function for this approach is shown in formula 1:

$$\text{Fitness} = \sum CF + \frac{1}{2} \left( \frac{\sum S\_cost}{\text{Total } S\_cost} + \frac{\sum S\_Vars}{1.5 \times \text{Vars Count}} \right) \quad (1)$$

In formula 1,  $CF$  shows conflict and  $\sum CF$  indicates total number of conflicts in the chromosome.  $\sum S\_Cost$  is the sum of spill costs of spilled variables and  $\text{Total } S\_cost$  is the sum of spill costs of all nodes.  $\sum S\_Vars$  shows the number of spilled variable and  $\text{VarsCount}$  indicates number of all graph nodes. The graph is considered to be colored correctly when  $\sum CF$  is equal to zero that means the graph is colored without any conflict in neighbor nodes.

The fitness function will be more than 1 only if there is a conflict in the graph. This property of fitness function is used for mutation operator.

Proposed algorithm chooses a chromosome with the minimum fitness value and if the value of fitness function is equal to zero it mean that algorithm colored the graph without any spill.

## 4.3 Initial Population

In order to generate an initial population, for ten times of existing nodes of interference graph, the chromosome is being produced.

The chromosomes are arrays with the length of the number of the nodes for each of the columns that shows a node of a graph the number between zero and the number of registers have been assigned. For generating better initial population, it has been used from some of the specification of nodes.

In the proposed solution the initial population is produced in three ways. The first way is randomly, in which 30 percent of the chromosomes are generated in this way, the rest of the chromosomes are produced, by  $S\text{-Degree}$  parameter which is shown in the two formulas in the below:

$$\begin{aligned} S\text{-Degree}(i)_1 &= S\text{-cost}(i) \\ S\text{-Degree}(i)_2 &= \frac{S\text{-cost}(i)}{\text{Degree}(i)^2} \end{aligned} \quad (2)$$

In the above formulas,  $S\text{-cost}(i)$  shows the spill cost of variable and  $\text{Degree}(i)$  shows the degree of the node. The function is in this way that;  $S\text{-Degree}$  are sorted decreasingly and assign zero color to the nodes which their  $S\text{-Degree}$  is less than others and their degrees be more than the number of registers.

#### 4.4 Selection Method

In order to select chromosome among the initial population the *tournament* is used that it chooses 4 chromosomes randomly and move that one with best fitness to intermediate population. This process is repeated until intermediate population is complete. Furthermore, selected chromosomes must not be omitted in the Initial population.

#### 4.5 Crossover and Mutation

In this approach, single-point crossover with the probability of 0.4 is used and a problem-specific mutation is introduced that has an important role in proposed approach. As mutation probability in proposed approach is more than usual and it emphasizes mutation it more toward the evolutionary strategy. Mutation probability is 0.9. The general function of mutation is in this way that it separates the chromosomes in two groups.

The first group is those with fitness function value more than 1, it means in these chromosomes there are conflicts. In this way the nodes must be spilled or the color

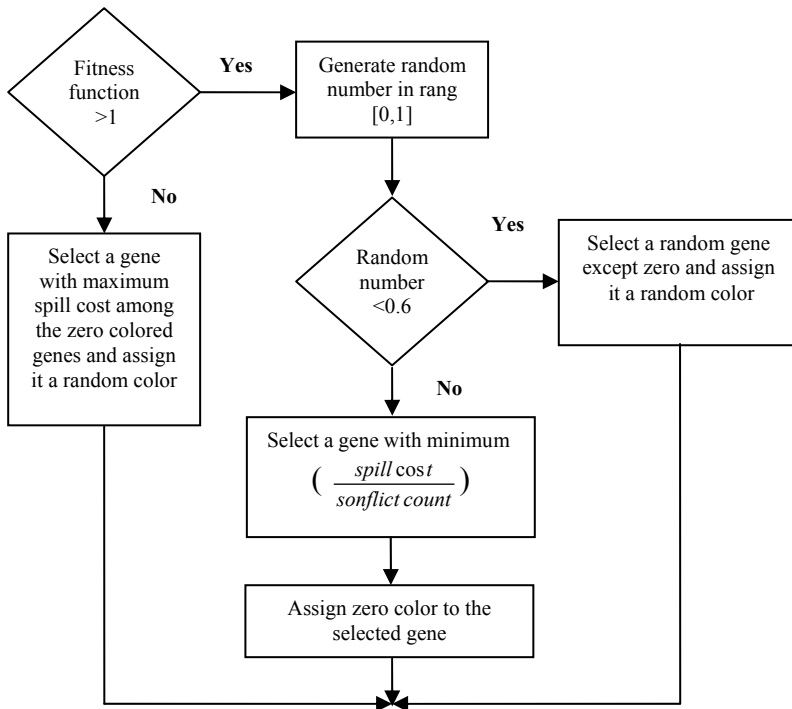


Fig. 3. Mutation operand

has to be changed for doing this in 60 percent of chromosomes, that gene is selected which this formula  $SpillCost / ConflictCount$  in it is the lowest and it takes zero color, or it will be spilled from the graph. In the rest of the 40 percent a randomly gene among other, except zero are selected and it's color changes.

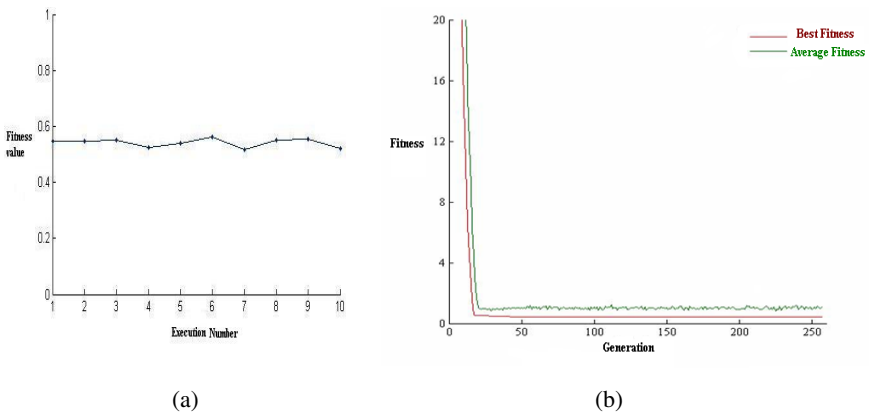
The second group is those chromosomes with their fitness function is less than 1, it means there isn't any conflict in these chromosomes. In this case for improving the chromosome, those genes which their color are zero must be lowered for doing this, among the genes which have zero color, the genes with the higher cost is chosen and another color is specified to it randomly. The stages of the mutation are shown in figure 3.

## 5 Evaluation and Experimental Results

The proposed method is implemented by visual basic language. By doing several experiments the best population size is 10 times of the number of nodes and the number of generation is considered 600. For showing the performance of algorithm, the graphs randomly produced, and the proposed algorithm is done on these graphs. For showing the durability of algorithm, there are some experiments and the result is shown on the fig. 4(a). RAGCEA algorithm executed 10 times on a sample graph with 100 nodes, 0.5 edge density and 10 color, then results are shown in the average of fitness value equal to 0.5415 and its variance is 0.00022.

In order to show convergence of RAGCEA algorithm it was executed on the graph with 100 nodes and 250 generations and in each generation average of fitness value and the best fitness value were calculated and the result, are shown in fig. 4(b). It is obvious that best fitness value in each generation is better than previous generations.

In order to generate random graphs, the method that Topcuglu and his colleagues used in their paper is used in this paper. For doing this the graph generator takes 3



**Fig. 4.** (a) Stability of RAGCEA algorithm. (b) Convergence of RAGCEA algorithm.

input parameters: The number of nodes ( $n$ ) in the graph, edge density ( $\alpha$ ) and the mean spill cost value ( $\gamma$ ) for the variables. The edge density is the probability with which an edge is present between any two nodes. The total number of edges in a randomly generated interference graph is close to  $\alpha \times (\frac{n \times (n-1)}{2})$ .

In this study experimental evaluation includes disconnected interference graphs as well. After a graph is constructed, the spill cost of each node is set randomly using a uniform distribution with the range of  $[1, \dots, 2\gamma]$ . Each generated graph is tested using various numbers of registers. For this purpose, the register density ( $\beta$ ) term is introduced in the experiments. Note that the total number of registers in the system is equal to  $r = \beta \times n$ ,  $r$  shows the number of registers. For each problem size, the edge density is varied using the values in the set  $\{0.05, 0.1, 0.25, 0.5, 0.75, 1.0\}$  and the register density ( $\beta$ ) is varied over the values from the set  $\{0.02, 0.04, 0.08, 0.1, 0.15, 0.2\}$ .

A set of 100 different interference graphs are generated with a fixed edge density value for each problem size; and each generated graph is tested using various number of registers by varying the register density. The total number of cases considered for each edge density is equal to 600, since there are 100 different random graphs for each six different register densities. The spill costs of variables are uniformly distributed across the range of  $[1, \dots, 10]$ .

In this paper we compare our approach with 3 different algorithms. Table 1 shows the results of execution of algorithm on the randomly generated graphs. Table 2 shows the comparison of these algorithms in some aspects.

**Table 1.** Comparison of algorithms with respect to total costs and number of spilled variables for various edge densities

n	$\alpha$	Spill Cost				Spilled variables			
		RAG CEA	HEA [9]	OCH [9]	GPX [9]	RAG CEA	HEA [9]	OCH [9]	GPX [9]
100	0.05	17.27	17.04	19.33	41.25	4.27	4.37	5.29	11.76
	0.1	40.22	41.15	49.65	87.33	9.38	9.38	11.92	21.92
	0.25	99.1	90.92	107.64	167.62	22.10	20.41	25.97	42.74
	0.5	185.76	167.44	192.96	254.33	42.55	37.50	45.39	59.36
	0.75	278.05	284.46	316.18	350.77	61.84	56.58	64.46	69.71
	1	446.85	456.16	456.16	456.16	90.20	90.19	90.19	90.19
200	0.05	10.95	12.10	18.31	108.97	3.50	3.9	6.46	31.27
	0.1	51.24	53.58	68.70	223.46	12.5	12.47	17.65	54.46
	0.25	169.30	148.91	171.33	393.23	37.93	33.62	41.81	94.28
	0.5	343.77	295.55	340.01	570.26	76.02	65	80.59	125.64
	0.75	536.3	504.46	543.32	713.73	115.73	104.20	120.17	145.73
	1	867.2	862.44	862.44	862.44	180.33	180.33	180.33	180.33

**Table 2.** Comparison of various Algorithms

	<b>GPX</b>	<b>OCH</b>	<b>HEA</b>	<b>RAGCEA</b>
Coloring and Spilling at the same time	×	×	✓	✓
Evolutionary Algorithm	✓	×	✓	✓
Graph Coloring	✓	✓	✓	✓

The results are showing that the proposed solution in this paper (RAGCEA) outperforms the OCH and GPX in all test cases and also it outperforms the HEA in some cases.

## 6 Conclusion

Register allocation has an important role in Compiler optimization phase. In this paper, RAGCEA method had been introduced for register allocation which we have used the evolutionary approach. In RAGCEA method, the allocation to registers and spilling the variables can be performed at the same time. We use the problem-specific mutation and this mutation operator has an important role in solving this problem.

The experimental evaluation based on randomly generated graphs reveals that our RAGCEA method outperforms a widely used register allocation heuristic (the OCH algorithm) and it outperforms HEA in some cases. It would be a good idea to implement this algorithm with parallel algorithms in order to improve its performance and also it can be improved by improving fitness function or mutation.

## References

1. Aho, A.V., Sethi, R., Ullman, J.D.: Compilers: Principles, Techniques, and Tools. Addison-Wesley, Reading (1986)
2. Beaty, S.J.: Register Allocation and Assignment in a Retargetable Microcode Compiler Using Graph Coloring. Colorado State University (1987)
3. Cooper, K.D., Dasgupta, A.: Tailoring Graph-Coloring Register Allocation for Runtime Compilation. IEEE Computer Society (2006)
4. Chaitin, G.: Register Allocation and Spilling via Graph Coloring. In: Sigplan 1982 (1982)
5. Chow, F.C., Hennessy, J.L.: The Priority-Based Coloring Approach to Register Allocation. ACM (1990)
6. Chaitin, G., Auslander, M., Chandra, A., Cocke, J., Hopkins, M., Markstein, P.: Register Allocation via Coloring. Computer Languages, 45–57 (1981)
7. Briggs, P.: Register allocation via graph coloring, Phd thesis, Rice University, Houston, USA (1992)
8. Wu, S., Li, S.: Extending Traditional Graph-Coloring Register Allocation exploiting Meta-heuristics for Embedded Systems. IEEE Computer Society (2007)
9. Topcuoglu, H.R., Demiroz, B., Kandemir, M.: Solving the Register Allocation Problem for Embedded Systems Using a Hybrid Evolutionary Algorithm. IEEE Transaction on Evolutionary Computation 11(5) (October 2007)
10. Hack, S., Goos, G.: Optimal Register Allocation for SSA-form programs in polynomial time. Information Processing Letters 98, 150–155 (2006)

# A Survey on Swarm and Evolutionary Algorithms for Web Mining Applications

Ashok Kumar Panda<sup>1</sup>, S.N. Dehuri<sup>2</sup>, M.R. Patra<sup>3</sup>, and Anirban Mitra<sup>1</sup>

<sup>1</sup>Department of CSE & IT, MITS, Rayagada, Orissa, India  
akpanda7@yahoo.co.in,  
mitra.anirban@gmail.com

<sup>2</sup>Department of Inf. & Comm. Technology, F.M. University, Balesore, Orissa, India  
satchi.lapa@gmail.com

<sup>3</sup>Department of Computer Science, Berhampur University, Orissa, India  
mrpatra12@gmail.com

**Abstract.** Internet is the biggest source of data and information today. It is the family of web sites and informative files. This paper focuses mainly on the web data and proposes some conceptual theories to extract knowledge through different web mining techniques like Clustering, FIS, ANN, LGP etc. We also focused on various aspects of applications of web mining in E-commerce & Business Intelligence. Finally, we discussed Swarm Intelligence (SI) techniques which are based on distributive self organized system such as Ant Colony Optimization (ACO), Stochastic Diffusion Search (SDS) and Particle Swarm Optimization (PSO) in brief in this survey which are preferred because of its vast uses and simplicity.

**Keywords:** Web mining, Clustering, E-commerce, Swarm Intelligence, Business Intelligence.

## 1 Introduction

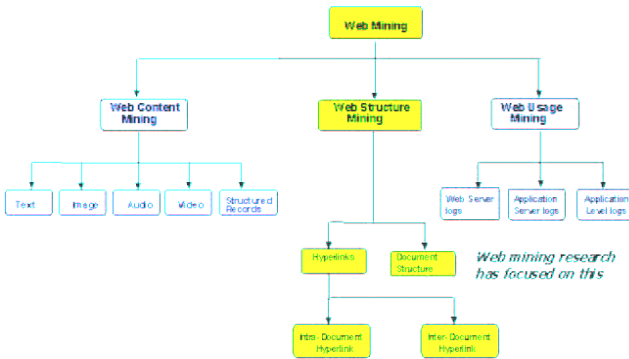
World Wide Web are serving us informations, in the order of hundreds of terabytes and is expanding rapidly. As the web is largely unrecognized with vast information, collecting and analyzing the data is very difficult. Web mining techniques are used to extract knowledge out of hundreds of web sites. Knowledge acquisition is used for activities in E-commerce. Accurate Web usage information helps customer management, improve cross marketing/sales, effectiveness of promotional campaigns, and find the most effective logical structure for the web space and so on. The rapid growth of E-commerce made Intelligent marketing to play a vital role for strategies and relationship management. Web mining attempts this & helps in effective web site management, creating adaptive web sites, business and support services, personalization, network traffic flow analysis and so on. Web mining extracts knowledge from web data - including web documents, hyperlinks, usage logs, etc . Here we followed the data-centric view for modifying the well existing definition is to add the extra clause about structure and usage data because mining web content

makes no difference whether the content was obtained from the Web, a database, a file system or through any other means.

## 2 Analysis and Definitions

### 2.1 Web Mining Taxonomy

Web Mining is divided into three distinct categories, according to the kinds of data to be mined. The Fig. 1[5] provides an overview of web mining taxonomy.



**Fig. 1.** Web Mining Taxonomy

*Web Content Mining* is the process of extracting useful information from the contents of Web documents. Content data corresponds to the collection of facts in web page. It consists of text, images, audio, video, or structured records such as lists and tables. Text mining and its applications have been the most widely researched topics.

*Web Structure Mining*, the structure of a typical web graph consists of web pages as nodes, and hyperlinks as edges connecting between two related pages. It is the process of discovering structure information from the Web. This is further divided into two kinds, based on the structural data used such as *Hyperlinks*, *Intra-Document Hyperlink* and *Document Structure*.

*Web Usage Mining* discovers interesting usage patterns from web data for analysis of customer profiles on web-based applications [SCDT2000]. Usage data captures the identity or origin of web users along with their browsing behavior in a web site. It is further classified depending on the kind of usage data i.e. *Web Server Data*, *Application Server Data* & *Application Level Data* [SCDT2000].

### 2.2 Web Mining in E-commerce

E-Commerce as an application of web mining determines lifetime value of clients, design cross marketing strategies, evaluate promotional campaigns, target electronic ads and coupons at user groups on their access patterns, predict user behavior from learned rules and analyze user profiling & presents dynamic information to users.



*Web usage mining Applications:* It has several applications in e-business, including personalization, traffic analysis, and targeted advertising. The development of graphical analysis tools such as Webviz[17] popularized web usage mining for web transactions. The main areas of applications is web log data preprocessing and identification of useful patterns from the preprocessed data. Also illustrated how ACO is linked with web usage mining[38] as application of swarm intelligence.

### 3 Business Intelligence

In this paper we discuss on how technologies for web data extraction, syndication and integration allow new applications and services in Business Intelligence (BI) and Semantic web domain . BI covers three main process steps: *data integration*, *data storage* and *data usage* . *Motivation* : Data available on the Web are crucial asset in enterprise world for making decisions on product & policies .Semantic web provides helpful means analyzing information on the market which is know as *competitive intelligence (CI)* or *competitive analysis (CA)*.

*Customer Profile - An Essential Tool* :The extension of BI on customer behavior and their access pattern in the web pages is termed as customer profiling & provides the most important informations for taking decision on the basic of their profiles. This focuses on customers' knowledge and ability to make informed decisions, and improve their lives. The Customer Profile requested information comprises of the following areas:

- Demographics
- Enterprises(s)
- Preferred methods of receiving information
- Business management practices
- Major sources of information used in management decisions.

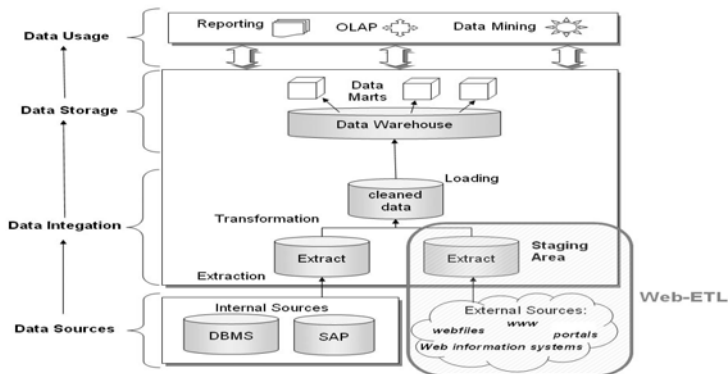


Fig. 2. The Business Intelligence reference process

*Semantic Web* termed as ‘an extension of the current web where information is given in well-defined meaning. The semi-structured web consisting billions of documents in different formats being not queriable as a database in heavily mix layout & structure, the intended information is presented. Until the current scenario is realized, “translation components” between the web, that (semi-)automatically translate web content (e.g.HTML) into a structured format(e.g.XML) are necessary. Once transformed it is used by applications stored into databases or populated into ontologies.

*Integrated Wrapper Technologies (IWT)*: A Wrapper is a program that automatically accesses source data (e.g. from the Web in HTML)then extracts and transforms it into another format (e.g. XML).IWT systems combine the capabilities of wrapping components with Information Integration (II) components [16]. It also extracts data from semi-structured web pages, transform into a semantically useful structure and then integrates with a Web ETL-process into a Business Intelligence system. In this work, the proposed architecture[17] of BI is dealt here with the three main components: *Business Data Definition, Customer Interaction, and Analysis* .

## 4 Algorithms /Approaches Discussed

Data mining techniques are very much used for extracting knowledge [33]from huge data. If it is from web data then web mining is considered. Web usage mining in particular has become very critical for effective web site management, creating adaptive web sites, business and support services, personalization, and network traffic flow analysis etc. Swarm intelligence techniques[25] proved better prediction in analysis of data in web usage mining. A novel approach called *intelligent-miner (i-Miner)*[2] which optimizesthe concurrent architecture of a fuzzy clustering algorithm (to discover web data clusters) and a fuzzy inference system to analyze the web site visitor -trends. The clustered data is then used to analyze the trends using a Takagi-Sugeno fuzzy inference system learned using a combination of evolutionary algorithm and neural network learning. Proposed approach is compared with self-organizing maps (to discover patterns) and several function approximation techniques like neural networks, linear genetic programming and Takagi-Sugeno fuzzy inference system (to analyze the clusters). In the following sub-sections, we discuss some of these algorithms.The different algorithms as depicted below, have been categorized under Three major approaches as Clustering , Computational Intelligence & Swarm Intelligence.

### 4.1 The Clustering Algorithms

*Fuzzy clustering-Means algorithm (FCM)*: This is one of the widely used clustering methods ,developed by Bezdek in 1981[2],[34].FCM partitions a collection of  $n$  vectors  $x_i, i = 1; 2 : : n$  into  $c$  fuzzy groups and finds a cluster center in each group such that a cost function of dissimilarity measure is minimized. To accommodate the introduction of fuzzy partitioning, the membership matrix  $U$  is allowed to have elements with values between 0 and 1.

*Self-organizing map (SOM)*: The SOM algorithm[2],[30] is used to cluster the user access records that visualize & interpret large high-dimensional data sets. The map consists of a regular grid of processing units, neurons. In a model of multidimensional

observation, a vector consisting of features, is associated with each unit. The map represents all the observations with optimal accuracy using restricted set of models. At the same time the models become ordered on the grid so that similar models gets close & dissimilar models stay far from each other. Fitting the model vectors is usually carried out by a sequential regression process, where  $t = 1; 2; \dots$  is the step index:

## 4.2 The Computational Intelligence(CI):

CI substitutes intensive computation for insight into how complicated systems work for trend analysis. Artificial neural networks, fuzzy inference systems, probabilistic computing, evolutionary computation(GP,LGP) etc are discussed under CI.

*Artificial Neural Network (ANN)*: ANNs were designed to mimic the characteristics of the biological neurons . Learning[32] occurs by example through training, where the training algorithm iteratively adjusts the connection weights .Back propagation is one of the most famous training algorithms for multilayer perceptions & a gradient descent technique to minimize the error E for a particular training pattern. [2].

*Linear Genetic Programming (LGP)*: Linear genetic programming proposed by Banzhaf et al.1998 [2] is a variant of the GP technique that acts on linear genomes. This can tremendously hasten up the evolution process as, no matter how an individual is initially represented, finally it has to be represented as a piece of machine code, where fitness evaluation requires physical execution of the individuals.

*Fuzzy inference systems (FIS)*: Fuzzy logic provides a framework to model uncertainty, human way of thinking, reasoning and the perception process. Fuzzy if-then rules & fuzzy reasoning are the backbone of FIS, which are the most important modeling tools based on fuzzy set theory. A Takagi Sugeno FIS [2],[26] has fuzzy rules, constituted by weighted linear combination of crisp inputs rather than fuzzy set.

## 4.3 The SWARM Intelligence Techniques

In broad sense, how system learns, many algorithms are available including Swarm Intelligence. Here population is made up of agents. These agents interact locally i.e. with each other and to the environment to find the solution but doesn't have any central authority to control them. So their interactions lead into global behavior of the system. This technique is also inspired by the elements of nature like teamwork of ants, birds flying together, animal moving in herd etc [15],[17].

Here we discuss Ant Colony Optimization (ACO), Stochastic Diffusion Search (SDS) and Particle Swarm Optimization (PSO) under SI. ACO introduced by Marco Dorigo [15] where each agent moves along the problem graph as artificial pheromone in such a way that future agent can build better solutions. SDS ,as a population-based, pattern-matching algorithm,(Bishop,1989)where each agent searches solution probabilistically and communicate hypothesis on one to one basis and the positive feedback system is so tuned that all the agents revolve around one global best solution that finds optimal solution. In PSO, each agent or particle is initially seeded into n-dimensional solution surface with initial velocity and a communication channel to other particles. Using some fitness function they are

evaluated after certain interval and particles are accelerated towards those particles which have higher fitness value. Being large numbers of particles in the populations it is less likely to converge in local minima showing more advantageous over other search algorithms [15].

*Ant Colony Optimization(ACO)* : ACO algorithm[37] is based on how ant forges its food. Initially ants wander randomly in search of food and as finds it returns to its colony leaving pheromones on its trail , and other ant follow this trail rather wandering. If the food is too far from the colony ,more is the time for evaporation of pheromones & if too near then takes less time to evaporate. Thus fewer ants will follow this trail and continue wandering around in search for food located closely to the colony. This indicates of a local minima allowing less ant to follow the trail [15],[36].However it doesn't produce optimal solution compared to other schemes but near optimal solution is guaranteed and is acceptable mostly.[17][18][19]. Another algorithm for mining classification rule, the Threshold Ant Colony Optimization Miner (TACO Miner) was proposed by K. Thangavel et al [39].

*Stochastic Diffusion Search(SDS)*: This technique is a two phase scheme, in the first phase all agents will explore search space randomly. All agents have atomic data unit (ADU) and when an agent hit solution i.e. it matches the ADU, it selects other agents randomly & communicate about its hit. This phase is diffusion phase. Whenever more number of agents points to same solution, search is terminated [15].The disadvantage of SDS is in search spaces, distorted heavily by noise. Diffusion of activity due to disturbances decreases an average number of inactive agents involved in random search & increases the time to reach the steady state [20].

*Particle Swarm Optimization(PSO)* : Particle Swarm Optimization[31] is modeled by particles in multidimensional space having position and velocity. These particles fly through hyperspace and remember the best position they have seen. Members of a swarm communicate good positions to each other and adjust their own position and velocity. Communication is done on the best known swarm to all and the local bests known in neighborhoods of particles .[27],[28],[29]. Position and velocity is updated at each iteration following the formula

$$\begin{aligned}x &\leftarrow x + v \\v &\leftarrow wv + c_1r_1(\hat{x} - x) + c_2r_2(\hat{x}_g - x)\end{aligned}$$

where 'w' is the inertial constant and slightly less than 1.  $c_1$  and  $c_2$  are constants that say how much the particle is directed towards good positions. Good values are usually right around 1.  $r_1$  and  $r_2$  are random values in the range [0,1] [15].  $\hat{x}$  is the best the particle has seen.  $\hat{x}_g$  is the global best seen by the swarm. This can be replaced by  $x_1$ , the local best, if neighborhoods are being used. It may be noted that, this scheme is useful for solution which can be broken into partial solution & each particle optimize the partial solution [21],[22]. Recent works also illustrated some recommender systems using PSO[35].

## 5 Conclusion

As the Web and its usage continues to grow, the opportunity to analyze web data and extract useful knowledge also increases. In this work, we discussed some basics on web mining techniques & applications to E-commerce, Business Intelligence & Customer Profiling. Then introduced some ongoing & past works on i-miner and swarm intelligence algorithms with emphasis on PSO & ACO on data mining linking web usage mining. However, Swarm Intelligence, provides a distributive approach with a very simple natural process of cooperation. Web informations in different formats, distributed over various web pages are divided into various clusters. Each cluster is analyzed using swarm intelligences techniques which proves better efficiency than other complex intelligence agent. Finally this survey proposes application of swarm intelligence methods that reduces both time and space complexity for extraction of knowledge from web data.

## References

1. Abraham, A.: i-Miner, a Web Usage mining framework using Hierarchical Intelligent Systems. In: IEEE International Conference on Fuzzy Systems, FUZZY-IEEE 2003, pp. 1129–1134 (2003)
2. Abraham, A.: Business Intelligence from Web Usage Mining. *Journal of Information & Knowledge Management* 2(4), 4375–4390 (2003)
3. Chi, E.H., Rosien, A., Heer, J.: Lumberjack: Intelligent Discovery and Analysis of Web User Traffic Composition. In: *Proceedings of ACM SIGKDD Workshop on Web Mining for Usage Patterns and User Profiles*. ACM Press, Canada (2002)
4. Kosala, R., Blockeel, H.: Web Mining research: A Survey. *ACM SIGKDD Explorations* 2(1), 1–15 (2002)
5. Etzioni, O.: The World Wide Web: Quagmire or Gold Mine? *Comm. ACM* 39(11), 65–68 (1996)
6. 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)
7. Chakrabarti, S., et al.: Mining Web's Link Structure. *Computer* 32(8), 60–67 (1999)
8. Kumar, R., et al.: Trawling the Web for Emerging Cyber communities. In: *Proc. 8th World Wide Web Conf.* Elsevier Science (1999)
9. Pitkow, J.E., Bharat, K.: WebViz: A Tool for WWW Access Log Analysis. In: *Proc. 1st Int'l Conf. World Wide Web*, pp. 271–277. Elsevier Science (1994)
10. Srivastava, J., et al.: Web Usage Mining: Discovery and Applications of Usage Patterns from Web Data. *ACM SIGKDD Explorations* 1(2), 12–23 (2000)
11. Punin, J., Krishnamoorthy, M.: Extensible Graph Markup & Modeling Language Specification (1999), <http://www.cs.rpi.edu/~puninj/XGML/draftxgml.html>
12. Punin, J., Krishnamoorthy, M.: Log Markup Language (LOGML) Specification (2000), <http://www.cs.rpi.edu/~puninj/LOGML/draft-logml.html>
13. Maler, E., De Rose, S.: XML Linking Language (1998), <http://www.w3.org/TR/WD-xlink>
14. Mannila, H., Toivonen, H., Verkamo, I.: Discovering frequent episodes in sequences. In: *1st Intl. Conf. Knowledge Discovery and Data Mining* (1995)

15. Advances in Web Usage Mining and User Profiling. In: Masand, B., Spiliopoulou, M. (eds.) WebKDD 1999. LNCS (LNAI), vol. 1836. Springer, Heidelberg (July 2000)
16. Mahat, P.: S I & Machine Learning. Res. Report, Dept. CS, LAMAR Univ.
17. Ansari, S., et al.: Integrating E-Commerce & data mining: Architecture & Challenges. In: WEBKDD 2000 Workshop (2000)
18. <http://www.wikipedia.org>
19. [http://en.wikipedia.org/wiki/Swarm\\_intelligence](http://en.wikipedia.org/wiki/Swarm_intelligence)
20. [http://en.wikipedia.org/wiki/Ant\\_colony\\_optimization](http://en.wikipedia.org/wiki/Ant_colony_optimization)
21. <http://www.codeproject.com/cpp/GeneticandAntAlgorithms.asp>
22. <http://www.aco-metaheuristic.org/>
23. [http://en.wikipedia.org/wiki/Stochastic\\_Diffusion\\_Search](http://en.wikipedia.org/wiki/Stochastic_Diffusion_Search)
24. [http://en.wikipedia.org/wiki/Particle\\_swarm\\_optimization](http://en.wikipedia.org/wiki/Particle_swarm_optimization)
25. Grosan, C., et al.: Swarm Intelligence in Data Mining. SCI 34 I-20-2006. Springer, Heidelberg (2006)
26. Chen, Y., Peng, L., Abraham, A.: Programming Hierarchical Takagi Sugeno Fuzzy Systems. In: 2nd International Symposium on Evolving Fuzzy Systems (EFS 2006). IEEE Press (2006)
27. Eberhart, R.C., Shi, Y.: Particle swarm optimization: developments, applications & resources. In: Proceedings of the IEEE Congress on Evolutionary Computation, CEC, Seoul (2001)
28. Hu, X., Shi, Y., Eberhart, R.C.: Recent Advances in Particle Swarm. In: Proceedings of Congress on evolutionary Computation (CEC), Portland, Oregon, pp. 90–97 (2004)
29. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In: Proceedings of IEEE International Conference on Neural Networks, Perth, Australia, vol. IV, pp. 1942–1948. IEEE Service Center, Piscataway (1995)
30. Merkl, D.: Text mining with self-organizing maps. In: Handbook of Data Mining and Knowledge, pp. 903–910. Oxford University Press, Inc., New York (2002)
31. Pomeroy, P.: An Introduction to Particle Swarm Optimization (2003), <http://www.adaptiveview.com/articles/ipsop1.html>
32. Settles, M., Rylander, B.: Neural network learning using particle swarm optimizers. In: Advances in Information Science and Soft Computing, pp. 224–226 (2002)
33. Sousa, T., Neves, A., Silva, A.: Swarm Optimisation as a New Tool for Data Mining. In: International Parallel and Distributed Processing Symposium (IPDPS 2003), p. 144b (2003)
34. Steinbach, M., Karypis, G., Kumar, V.: A Comparison of Document Clustering Techniques. In: Text Mining Workshop, KDD (2000)
35. Ujgin, S., Bentley, P.J.: Particle swarm optimization recommender system. In: Proceedings of the IEEE Swarm Intelligence Symposium (SIS 2003), Indianapolis, Indiana, USA, pp. 124–131 (2003)
36. Weng, S.S., Liu, Y.H.: Mining time series data for segmentation by using Ant Colony Optimization. European Journal of Operational Research (2006), <http://dx.doi.org/10.1016/j.ejor.2005.09.001>
37. Dorigo, M., Bonabeau, E., Theraulaz, G.: Ant algorithms and stigmergy. Future Generation Computer Systems 16, 851–871 (2000)
38. Abraham, A., Ramos, V.: Web Usage Mining Using Artificial Ant Colony Clustering and Genetic Programming. In: IEEE Congress on Evolutionary Computation (CEC 2003), pp. 1384–1391. IEEE Press, Australia (2003) ISBN 0780378040
39. Thangavel, K., Jaganathan, P.: Rule Mining Algorithm with a New Ant Colony Optimization Algorithm. In: Proc. of the International Conference on Computational Intelligence & Multimedia Applications, December 3-15, vol. 2, pp. 135–140 (2007)

# Exploration Strategies for Learning in Multi-agent Foraging

Yogeswaran Mohan and S.G. Ponnambalam

School of Engineering,  
Monash University, Sunway Campus,  
46150 Petaling Jaya, Selangor, Malaysia

**Abstract.** During the learning process, every agent's action affects the interaction with the environment based on the agent's current knowledge and future knowledge. The agent must therefore have to choose between exploiting its current knowledge or exploring other alternatives to improve its knowledge for better decisions in the future. This paper presents critical analysis on a number of exploration strategies reported in the open literatures. Exploration strategies namely random search, greedy,  $\epsilon$ -greedy, Boltzmann Distribution (BD), Simulated Annealing (SA), Probability Matching (PM) and Optimistic Initial Values (OIV) are implemented to study on their performances on a multi-agent foraging task modeled.

**Keywords:** Foraging-task, reinforcement learning, exploration strategies, learning policies, Q-Learning.

## 1 Introduction

In the reinforcement learning (RL) problem [8], an agent acts in an unknown environment or incompletely known environment with the goal of maximizing an external reward signal. Exploration versus exploitation (EvE) is one of the elementary issues in RL which has received significant attention among the researchers for last few decades. Hence, there have been a great deal of theoretical work published to address the matter. During the learning process, the agents must often choose between actions that appear to be optimal in order to achieve highest possible rewards during the agent's learning period. In order to do so, the agents must decide whether to explore to gain new information or to exploit based on the past experiences to maximize the reward [7]. Various exploration strategies have been introduced to address the balancing issue in EvE.

In this paper, a multi-agent foraging task have been modeled as a standard platform to test the exploration strategies reported in the open literature. Exploration strategies namely random search,  $\epsilon$ -greedy, Boltzmann Distribution (BD), Simulated Annealing (SA), Probability Matching (PM) and Optimistic Initial Values (OIV) are the widely used strategies which will be adopted to study on their performances in the multi-agent foraging task modeled. Foraging task has been chosen as the test bed because it is a task that resembles most of

the real-world applications such as hazardous waste cleanup, urban search and rescue, surveillance systems, planet exploration and more. In the foraging task modeled, there are 5 agents (Khepera II model) and 10 pucks which are placed at predetermined location within a 2000mm by 2000mm environment developed using the Webots [10] platform. The agents will wander in the environment to search and retrieve pucks back to the home location.  $Q$ -learning, which is a form of reinforcement learning will be used to tackle the foraging task modeled.

The performances of the strategies will be evaluated based on the amount of the pucks collected, number of collisions in the environment and the total time consumed to collect the pucks. These objectives will expose the advantages as well as the disadvantages of the strategies in optimizing the multi-agent foraging task modeled.

## 2 Related Literature

In this section, a short review about some of the widely used exploration strategies mentioned in the open literature which were adopted for the multi-agent foraging task modeled.

### 2.1 Random Search

Although random search is a very primitive method of exploration which is not widely used in the current stage of development in RL, the strategy is also included in this paper as a benchmark reference for the other exploration strategies. An agent under random search strategy will select a random action  $a \in A$  without any influence of the rewards that are observed from the environment. This makes the agents to purely explore around the environment with no proper directions towards the desired objectives.

### 2.2 Greedy

Greedy approach is the commonly used exploration strategy that is associated with standard  $Q$ -learning. Implementation of greedy can be seen in [6], where the authors described a formal and principled approach to imitation called implicit imitation between two agents.

$$a = \operatorname{argmax}_a Q_{(s,a)} \quad (1)$$

Following the greedy policy, the agent selects action  $a \in A$  based on the highest  $Q(s, a)$  estimate of the available actions. The action  $a$  considered by the greedy policy may depend on actions made so far but not on future actions.

### 2.3 $\epsilon$ -Greedy

The most popular exploration strategy is  $\epsilon$ -greedy [9]. Whiteson et. al. [11] have applied this exploration strategy in two different tasks specifically mountain



car task and server job scheduling and compared its performance with other exploration strategies.

$$a = \begin{cases} \text{rand}(a_n) & \text{rand}(0, 1) \leq \xi \\ \text{argmax}_a Q(s, a) & \text{otherwise} \end{cases} \quad (2)$$

$\epsilon$ -greedy allows a certain degree exploration with a probability of  $\epsilon$  where  $\epsilon$  is a small positive value,  $0 < \epsilon < 1$ . High values of  $\epsilon$  will force the agent to explore more frequently and as a result will prevent the agent from concentrating its choices to the optimal action, while giving the agent the ability to react rapidly to changes that takes place in the environment. Low value of  $\epsilon$  will drive the agent to exploit more optimal actions. For the studied case,  $\epsilon=0.2$  improved the quality of the solution marginally out of the set of tested  $\epsilon = \{0.1, 0.2, 0.3, 0.4, 0.5\}$  values.

## 2.4 Boltzmann Distribution

BD is a strategy that reduces the tendency for exploration with time. It is based on the assumption that the current model improves as learning progresses. BD assigns a positive probability to any possible action according to its expected utility and according to a parameter  $T$  called temperature [1]. Morihito et. al. [5] implemented BD for a new multi-agents flocking framework using  $Q$ -learning. BD assigns a positive probability for any possible action  $a \in A$  using [3].

$$P(a|s) = \frac{e^{(Q(s,a)/T)}}{\sum_b e^{(Q(s,b)/T)}} \quad (3)$$

$$T_{new} = e^{(-dj)} T_{max} + 1 \quad (4)$$

Actions with high  $Q(s, a)$  are associated with higher probability  $P_i$ .  $T$  decreases as iteration  $j$  increases over time. Therefore, as learning progresses, the exploration tendency of the agents reduces and BD strategy will tend to exploit actions with high  $Q(s, a)$ . For the foraging task in this paper, the parameter  $T_{max}$  is adjusted to 500, decay rate  $d$  is set 0.009.

## 2.5 Simulated Annealing

The SA exploration strategy was introduced by Guo et. al [3] where the paper reports experimental results on a 22x17 puzzle problem and comparing the capabilities of the different exploration strategies.

$$a = \begin{cases} \text{rand}(a_n) & \xi < e^{(Q(s, \text{rand}(a_n)) - \text{argmax}_a Q(s, a))/T} \\ \text{argmax}_a Q(s, a) & \text{otherwise} \end{cases} \quad (5)$$

Following this policy, the agent selects an arbitrary action  $a \in A$  and executes the action based on the probability defined in (5) where  $\xi \in (0, 1)$ . Otherwise the agent selects and executes the optimal action.  $T$  is a positive parameter called the temperature which is set through trial and error. At the beginning of the iteration, when the temperature is high, the SA algorithm policy allows more exploration. As the temperature drops based on (4), the SA algorithm policy reduces the exploration rate and drives the agent to exploit more optimal solutions. In this paper, parameter  $T$  is also adjusted to 500.

## 2.6 Probability Matching

One of most recent work was by Koulouriotis and Xanthopoulos [4] where the paper examines multi-armed bandit tasks to solve the EvE problem using PM and compared several other exploration strategies.

$$P_{max} = 1 - (K - 1)P_{min}, P_{min} \in (0, 1) \quad (6)$$

$$P_{a_i} = P_{min} + (1 - K.P_{min}) \frac{Q(s, a_i)}{\sum_{n=1}^K Q(s, a_n)} \quad (7)$$

The PM rule computes each action's selection probability  $P_{a_i}$  as the proportion of the action's  $Q(s, a_i)$  to the sum of all  $Q(s, a_n)$  estimates. To enforce a sufficient amount of exploration, the actions exploration rate above a threshold of  $P_{min}$ . Otherwise, an action which is inefficient in the early iterations would never be considered. The exploration rate for each action is also kept below  $P_{max}$  being  $K$  the number of actions available in the observed state  $s \in S$ .  $P_{min}$  is set to 0.1 in this experiment.

## 2.7 Optimistic Initial Values

Using OIV exploration strategy, the initial  $Q(s, a_i)$  value of each state action pair can be set to some overwhelmingly high number. If a state  $s$  is visited often, then its estimated value will become more exact. Thus, the agent will try to reach the more rarely visited areas, where the estimated state values are still high. A work by Even-Dar and Mansour [2] gave theoretical justification for the method. In their paper, they proved that if the optimistic initial values are sufficiently high, q-learning converges to a near-optimal solution. In our experiment, the initial  $Q(s, a_i)$  values are set to +20 to drive the agents to explore non-visited states.

## 3 Q-Learning

Q-learning enables the agent to compute the optimal action-value function by a direct interaction with the environment. The performance of Q-learning, is strongly influenced by the way the agent balances the EvE problem. At each time instant or episode, the agent has to exploit what it has already learned in order to obtain high rewards, but at the same time it also has to explore the environment in order to discover unexplored and potentially more rewarding states of the state space  $S$ . When the agent takes action  $a \in A$  in state  $s \in S$ , receives a reward  $r \in R$  and gets to new state  $s'$ , the estimation of the action-value function is updated as:

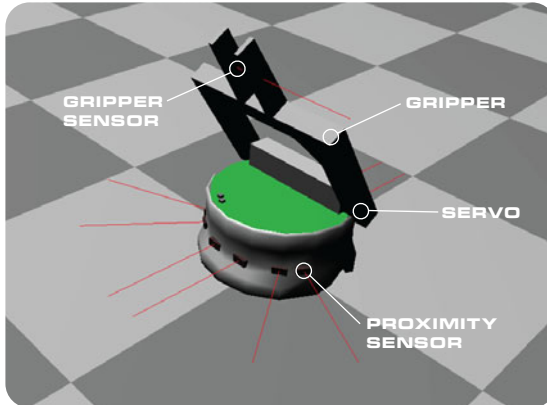
$$Q_{t+1}(s_t, a_t) = (1 - \alpha) Q_t(s_t, a_t) + \alpha [r_t + \gamma \max_b (Q_t(s_{t+1}, b))] \quad (8)$$

where  $0 \leq \alpha \leq 1$  is the learning rate and  $0 \leq \gamma \leq 1$  is the discount rate. The learning rate weighs the influence of the received rewards in the learning

process. A learning rate of 0 will make the agents not learn anything, while a rate of 1 would make the agents consider only the most recent reward. The discount factor weighs the influence of the future rewards. A rate of 0 will make the agents opportunistic by only considering current rewards, while a rate approaching 1 will make it venture for a long-term high reward.

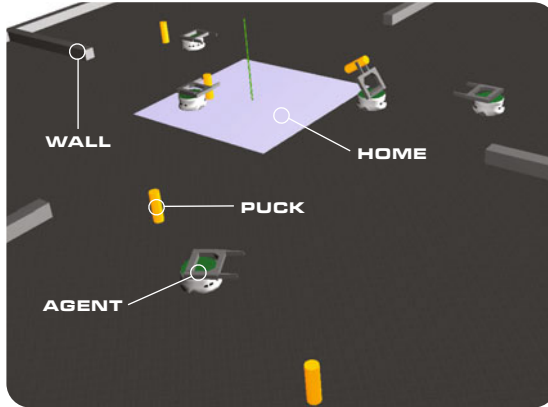
## 4 The Foraging Task

In this section, the descriptions of the model for the foraging task implemented in this paper is discussed. Fig.1 shows the model of the agent modeled for the experiment. The agent is modeled strictly based on Khepera II mobile robot. Each agent consists of 8 proximity sensors. The maximum sensing range of the proximity sensors are set to 50 mm and the minimum range to 20 mm. When an obstacle enters the minimum sensing range of the agent, the agent is considered to have collided with an obstacle. There is a proximity sensor used to sense the presence of puck in the gripper module. Based on the input from the proximity sensor, the agent will react accordingly by gripping or releasing the puck from or to the desired location. The simulation model can be directly downloaded into a real Khepera II mobile robot from Webots for real time experiments.



**Fig. 1.** The RL agent (Khepera II model). The agent in the simulation is modeled as similar as possible to the real physical mobile-robot.

Fig.2 shows the dimension and layout of the environment. The width and length of the environment each are respectively 2000mm. The environment is separated into grid of 100 by 100 mm for the agent to explore and exploit. Walls are located in the environment to divide the environment into four rooms and also act as a boundary for the environment. The walls also acts as obstacles for the agents. The home location is located exactly in the middle of the environment. The agents will pick up the pucks from the environment and deposit the



**Fig. 2.** The model of the environment

collected pucks at the home position. Pucks and agents are placed in the environment at predetermined locations at the beginning of the simulation. There are 10 pucks and 5 agents distributed evenly in the environment.

Set of states  $S$  observed by the agents are of  $\{x_i, z_i, p\}$  where  $x_i$  is the coordinate location in the x-axis,  $z_i$  is the coordinate location in the z-axis and  $p$  is the variable representing the presence of puck in the gripper module of the agent.  $p$  is set to 0 if there is no puck in the gripper module and set to 1 otherwise.

Based on the policy adopted the agents will choose the desirable action  $a \in A$  at the current available states  $s \in S$ . The set of actions changes relatively to the agents. The x-axis and z-axis has a coordinate value ranging from -1000 to +1000. Including the puck availability  $p$ , there are 882 states overall for the agents to explore and exploit in the foraging task modeled.  $Q(s, a)$  values that are updated or reinforced are stored into a table called  $Q$ -table.

Rewards show the desirability of the action taken by the agents towards the perceived states. If the action taken is attractive, a positive reward is given to the agent. If the action taken is not attractive then a negative reward is given to the agent. A +20 reward is given to the agents for picking up a puck from the environment. This triggers the agents to visit the same location again to look for more rewards. A reward of +20 is also given to the agents for dropping the puck at home location. A punishment of -1 is given if the agents experience any collision with the obstacles in the environment.

## 5 Experimental Procedures

The agents are independent learners, meaning each agent maintains individual  $Q$ -table to be updated using (8). The agents also do not communicate directly with each other. The corresponding coordinates of the agents are submitted to a central unit called the supervisor, and the supervisor will compile the coordinates of the agents and sends the information back to the agents regarding

the coordinate location of the other agents in the environment. If the agents are within the maximum sensing range defined earlier in Fig. 1 the agents will take action with the highest distance with the neighboring agents. The actions taken this way are not updated in  $Q$ -table.

The simulation starts by setting the episode counter, timer and  $Q(s, a)$  values in the  $Q$ -table are set to 0. In the initialization stage, the timer count is set to 0 followed by the resetting of agent's and the puck's locations to their respective coordinates in the environment. At the execution of the simulation, the agent observes the available states  $S$  from current state  $s_c$ . For each current state  $s_c$ , there are four available states and four available actions  $A = \langle north, south, east, west \rangle$  leading to the observed states. Based on the adopted learning policy, the agent selects the desired action and moves to the observed state  $s_i \in S$ . A reward  $r_i \in R$  is given to the agent based on the attractiveness of the action taken from the observed states. The relevant  $Q(s, a)$  value in the  $Q$ -table is updated using (8).

If the agent finds a puck, the agent will grab the puck and a reward of +20 will be given to the agent. This reward will be updated in the  $Q$ -table. After the update, the agent will be performing homing until the puck is deposited at the home location. A reward of +20 is also given if the agent successfully deposits the puck at the home location. If the agents encounter collision in the environment, a reward of -1 is given to the agent. After the related  $Q(s, a)$  values have been updated, the agent will move back to its current state and actions will be taken again after evaluation the observed current state  $s$ . These process repeats until termination conditions are met. At current state  $s$ , the agents will observe the available states  $S$ . The agent will then select an action  $a \in A$  based on the policy adopted. Reward  $r \in R$  for the action  $a \in A$  taken for state  $s \in S$  is calculated. Then the current state  $s$  is set to new state  $s'$ . The episode counter is incremented by 1.

$$R_t = (1 - [Timetaken/Totalsimulationtime]) \quad (9)$$

If 10 hours of the simulation time is complete or the number of pucks collected are 10, the simulation will be reseted again. This informs the completion of one run. At the end of each simulation run completion, a single reward value will be added to the overall rewards collected by the agents using (9) to include the reward for the time achievements of the adopted exploration strategy. At the beginning of each run, the timer will be set to 0, pucks and agents will be placed again to the predetermined location in the environment. Else the agents will continue to map the state-action pairs until the termination conditions are met. The simulation is reseted for 30 runs for each exploration strategy and the simulation commences until 10 sets of 30 runs are completed for each exploration strategy. At the beginning of each set, the initialization stage is carried out again. Finally the average data acquired for 30 runs are presented and discussed.

## 6 Results and Discussion

The performances of the exploration strategies in the experiment conducted are reported in this section. The simulation was carried out using Webots (10)

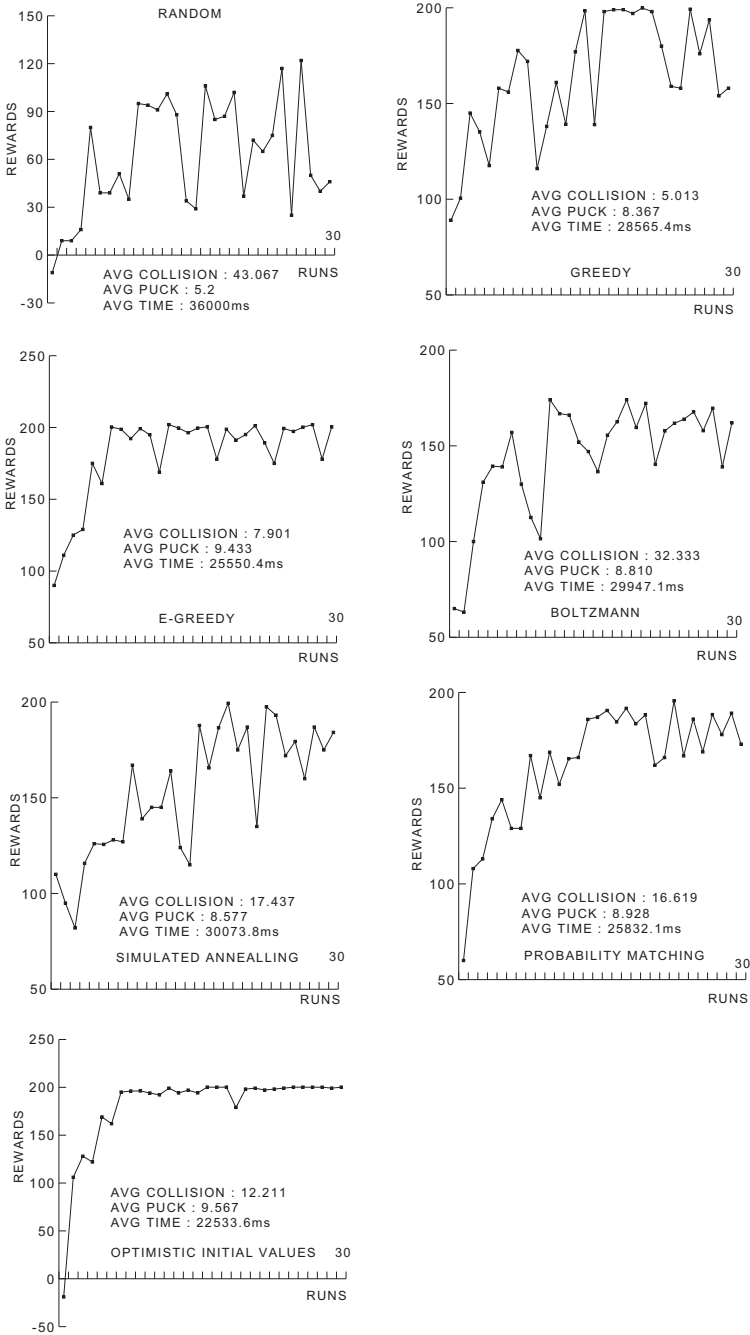


Fig. 3. Experimental results

platform. Each of the exploration strategies are tested for 10 sets of 30 runs and the average results are presented in Fig. 3. It is observed that greedy exploration strategy outperformed other exploration strategies by achieving an average of only 5.013 collisions throughout the entire simulation period. OIV exploration strategy outperformed other exploration strategies by collecting an average of 9.567 pucks throughout the entire simulation period. The OIV exploration strategy also outperformed in collecting and depositing the pucks within a average time of 22533.6ms. Based on the data, we can conclude that OIV performs well overall compared to the other strategies studied for the multi-agent foraging task modeled in this paper.

The characteristics of the greedy strategy prevents exploitation of undesired states as the learning process converges made it suitable for the collision avoidance in the foraging task modeled. However the performance of the greedy strategy is very poor in other objectives due to lack of exploration in the environment. Although  $\epsilon$ -greedy strategy promises convergence, it displays a number weaknesses. The agent under this strategy have no control over when or how to explore. Therefore, the agent may chose to take a random action when an obvious action might lead to a better reward and chose an optimal action without knowing whether the taken action is appropriate or not. Although BD and SA allows high exploration in the early stage of the learning process, both strategies generally requires a high scale of exploration time before converging. Therefore both BD and SA are not suitable for tasks that are constrained by time factors. PM on the other hand provides a probability for all the actions to be chosen. Although action yielding high rewards have higher probability of being chosen, the selection is still random. OIV is a simple method of biasing the initial  $Q(s, a)$  values to a very high value. Therefore, unexplored actions have greater  $Q(s, a)$  value estimates than explored ones, so unexplored actions tend to be selected. When all actions have been explored a sufficient number of times, the true  $Q(s, a)$  value function overrides the initial  $Q(s, a)$  value function estimates. This ensures the exploration is done to all the available states before making a valid judgment or decision in the action selection.

## 7 Conclusions

In this paper, the behavior of a  $Q$ -learning agents using the reported exploration strategies namely random search, greedy policy,  $\epsilon$ -greedy policy, BD, SA, PM and OIV have been studied in a foraging task and the results are reported. Through the experiment conducted, although all the strategies carry their own advantages and disadvantages, it is clear that OIV exploration strategy is much more practical and effective compared to the other strategies studied for the multi-agent foraging task studied. Future works should target more recent exploration strategies such as R-Max and model-based interval exploration (MBIE).

**Acknowledgment.** This research is funded by e-Science grant provided by Ministry of Science, Technology and Innovation (MOSTI), Malaysia, Project Number: 03-02-10-SF0036.

## References

1. Carmel, D., Markovitch, S.: Exploration strategies for model-based learning in multi-agent systems: Exploration strategies. *Autonomous Agents and Multi-agent Systems* 2(2), 141–172 (1999)
2. Even-Dar, E., Mansour, Y.: Convergence of optimistic and incremental Q-learning. *Advances in Neural Information Processing Systems* 2, 1499–1506 (2002)
3. Guo, M., Liu, Y., Malec, J.: A new Q-learning algorithm based on the metropolis criterion. *IEEE Transactions On Systems, Man, And Cybernetics? Part B: Cybernetics* 34(5), 2141 (2004)
4. Koulouriotis, D.E., Xanthopoulos, A.: Reinforcement learning and evolutionary algorithms for non-stationary multi-armed bandit problems. *Applied Mathematics and Computation* 196(2), 913–922 (2008)
5. Morihiro, K., Isokawa, T., Nishimura, H., Matsui, N.: Emergence of Flocking Behavior Based on Reinforcement Learning. In: Gabrys, B., Howlett, R.J., Jain, L.C. (eds.) *KES 2006. LNCS (LNAI)*, vol. 4253, pp. 699–706. Springer, Heidelberg (2006)
6. Price, B., Boutillier, C.: Accelerating reinforcement learning through implicit imitation. *Journal of Artificial Intelligence Research* 19(1), 569–629 (2003)
7. Strehl, A., Li, L., Wiewiora, E., Langford, J., Littman, M.: PAC model-free reinforcement learning. In: *Proceedings of the 23rd International Conference on Machine Learning*, p. 888. ACM (2006)
8. Sutton, R., Barto, A.: *Reinforcement learning: An introduction*. The MIT Press (1998)
9. Szita, I., Lőrincz, A.: The many faces of optimism: a unifying approach. In: *Proceedings of the 25th International Conference on Machine Learning*, pp. 1048–1055. ACM (2008)
10. Webots: <http://www.cyberbotics.com>, <http://www.cyberbotics.com>, commercial Mobile Robot Simulation Software
11. Whiteson, S., Taylor, M., Stone, P.: Empirical studies in action selection with reinforcement learning. *Adaptive Behavior* 15(1), 33 (2007)



# Nurse Rostering Using Modified Harmony Search Algorithm

Mohammed A. Awadallah<sup>1</sup>, Ahamad Tajudin Khader<sup>1</sup>,  
Mohammed Azmi Al-Betar<sup>1,2</sup>, and Asaju La'aro Bolaji<sup>1</sup>

<sup>1</sup> School of Computer Sciences, Universiti Sains Malaysia (USM), 11800,  
Pulau Pinang, Malaysia

<sup>2</sup> Department of Computer Science, Al-zaytoonah University,  
Amman, Jordan

{mama10\_com018, ab110\_sa0739}@student.usm.my,  
{tajudin, mohbetar}@cs.usm.my

**Abstract.** In this paper, a Harmony Search Algorithm (HSA) is adapted for Nurse Rostering Problem (NRP). HSA is a global optimization method derived from a musical improvisation process which has been successfully tailored for several optimization domains. NRP is a hard combinatorial scheduling problem of assigning given shifts to given nurses. Using a dataset established by International Nurse Rostering Competition 2010 of sprint dataset that has 10-early, 10-late, 10-hidden, and 3-hint. The proposed method achieved competitively comparable results.

## 1 Introduction

The Nurse Rostering Problem (NRP) is normally tackled by assigning a set of shifts of different types to a limited number of nurses with different working skills and working contracts. Four main factors should be considered during the construction of a nurse roster: hospital management policies, government regulations, fair shift distribution among nurses, and nurses' preferences [1]. Basically, these factors are classified into hard and soft constraints based on the hospital administrative perspective. Hard constraints are those that should be satisfied, while violations of the soft constraints are allowed but should be avoided, if possible. From hospital administrative perspectives, the difficulties of constructing the nurse roster arise due to a considerable number of constraints which vary amongst hospitals. A nurse roster can be said to be *feasible* by satisfying hard constraints but its quality is determined by satisfying soft constraints. However, it is almost impossible to find a roster that fulfills all constraints. Computationally, this is a combinatorial optimization problem, which belongs to an NP-complete class in almost all its variations [2]. NRP is a hard scheduling problem which does not lend itself to be solved by manual methods.

In the past, the head nurse used to generate a nurse roster based on accumulated personal experience by taking into account the set of constraints to enhance the quality of the nurse roster. This would consume considerable time without even meeting most constraints. Hence, operational researchers and artificial intelligence

experts have been directing their attention to solve NRP. Different approaches have therefore been proposed, such as Tabu Search [3], Variable Neighbourhood Search [4][5], Simulated Annealing [6], Ant Colony Optimization [7], Genetic Algorithm [8], Electromagnetic Algorithm [9], Scatter Search [10], Memetic Algorithm [11]. More information about these and other methods can be seen in the surveys [1][12].

Harmony Search Algorithm (HSA) is a new population-based metaheuristic proposed by Geem et al., [13]. It has been successfully applied to a wide variety of optimization problems such as structural design [14], vehicle routing [15], water network design [16], tour routing [17], traveling salesman problem [13], course timetabling [18][19], examination timetabling [20], and many others as reported in [21][22]. HSA has different characteristics: (i) it is easy to tailor for different types of optimization problems; (ii) it requires few mathematical requirements and does not require a derived value of the decision variables in the initial stage of search [23]. Therefore, the performance is improved by tuning HS parameters [24][25], hybrid with other methods such as particle swarm optimization [26][27][28], and ant colony algorithm [29].

HSA is derived from the behavioral phenomenon of musicians in the improvisation process, where a set of musicians play pitches of their instruments repetitively to come up with pleasing harmony as determined by an aesthetic standard value. Analogously in optimization, a set of decision variables is assigned with values, iteratively to come up with a (near) optimal solution as determined by the objective function. Practically, the HSA is an iterative process that starts with a set of initial solutions stored in harmony memory (HM). At each iteration, a new solution (new harmony) is generated and evaluated to replace the worst solution in the HM, if it is better. This process is repeated until a stop criterion is met.

The main objective of this paper is to alert HSA for NRP as an initial exploration to investigate the effectiveness of such method in the nurse rostering domain, henceforth called Modified Harmony Search Algorithm (MHSA).

For evaluation purposes, a dataset established by the organizers of International Nurse Rostering Competition 2010 (INRC2010) is used. Note that this is the first standard dataset for NRP with its instances divided into three categories: sprint, middle, and long distance. These are different in terms of size and complexity. Each category is further classified into four types: early, hidden, late, and hint. More details can be found in INRC2010 website<sup>1</sup>. In a nutshell, MHSA has been able to yield comparable results.

The outline of the paper is as follows: In Sect. 2, we describe the nurse rostering problem we were dealing with. In Sect. 3 and 4 we discuss the algorithm and results respectively. In Section 5 we draw conclusions on the success of this approach and present some possible future extensions.

## 2 Problem Description

NRP consists of a set of nurses to be assigned to a set of different shifts on daily basis over given time periods. Each nurse has a specific title with skills (i.e., Head Nurse,

---

<sup>1</sup> <http://www.kuleuvenkortrijk.be/nrpcpetition>

Nurse) determined by qualification and experience. Furthermore, each nurse is employed through a contract agreed upon with the hospital administration (i.e., Full time, Half time). These contracts that determine the nurse job specifications as formulated in the soft constraints ( $S_1, \dots, S_6, S_9, S_{10}$ ) are shown in Table 1.

**Table 1.** Description of INRC2010 Hard ( $H_1, H_2$ ) and Soft ( $S_1, \dots, S_{10}$ ) constraints

Symbol	The constraint
$H_1$	All demanded shifts must be assigned to a nurse.
$H_2$	A nurse can only work one shift per day, i.e. no two shifts can be assigned to the same nurse on a day.
$S_1$	Maximum and minimum number of assignments for each nurse during the scheduling period.
$S_2$	Maximum and minimum number of consecutive working days.
$S_3$	Maximum and minimum number of consecutive free days.
$S_4$	Assign complete weekends.
$S_5$	Assign identical complete weekends.
$S_6$	Two free days after a night shift.
$S_7$	Requested day-on/off.
$S_8$	Requested shift-on/off.
$S_9$	Alternative skill.
$S_{10}$	Unwanted patterns. (Where pattern is a set of legal shifts defined in terms of work to be done during the shifts [30]).

Table 2 gives detailed explanations for each shift which consists of the skill of the nurse, start and end time, and the number of nurses required for each day (i.e., shift demand).

**Table 2.** Shift Categories Details

Shift Details				Shift Demand							
Shift	Skill	Start Time	End Time	Mon	Tue	Wed	Thu	Fri	Sat	Sun	
D	Day Shift	Nurse	08:30	16:30	3	3	3	3	3	2	2
L	Late Shift	Nurse	14:30	22:30	9	9	9	9	9	5	5
E	Early Shift	Nurse	06:30	14:30	9	9	9	9	9	5	5
N	Night Shift	Nurse	22:30	06:30	3	3	3	3	3	2	2
DH	Head Nurse Day Shift	Head Nurse	08:30	16:30	2	2	2	2	2	1	1

It is worth mentioning that the nurse roster should satisfy the nurse preferences (see  $S_7, S_8$  in Table 1), for example the nurse preferences Day-OFF/ON (i.e., the nurse prefers (not) to work on a specific day) or Shift-OFF/ON (i.e., the nurse prefers (not) to be assigned to a specific shift on a specific day). Data for such preferences are gathered from the nurses well before handling the scheduling process.

Conventionally, the constraints in NRP are divided into two types: hard and soft constraints. Hard constraints ( $H_1, H_2$ ) are those that should be satisfied, while the fulfillment of the soft constraints ( $S_1, \dots, S_{10}$ ) is desired but not absolutely essential. The basic objective is to find a roster that satisfies all hard constraints while minimizing the penalty of soft constraint violations. The mathematical formulation of the two hard constraints is as follows:

**H<sub>1</sub>**: All demanded shifts must be assigned to a nurse (see (1)).

$$\sum_{i=1}^N x_i = d_{jk}. \tag{1}$$

**H<sub>2</sub>**: A nurse can only work one shift per day (see (2)).

$$\sum_{i=1}^N x_i \leq 1. \tag{2}$$

Where  $x_i$  is allocation in nurse roster solution ( $\mathbf{x}$ ) assigned with a triple of items (nurse  $u$ , day  $v$ , shift  $r$ ).  $d_{jk}$  is the number of nurses required for day ( $j$ ) at shift ( $k$ ). Note that,  $v = j, r = k$ , and  $N$  is the maximum length of allocations for solution roster ( $\mathbf{x}$ ) calculated as in (3).

$$N = \sum_{i=0}^{W-1} \sum_{j=1}^7 \sum_{k=1}^T d_{((i \times 7) + j)k}. \tag{3}$$

In (3). ( $W$ ) is the maximum number of weeks in a scheduling period, while ( $T$ ) is the total number of shifts.

The nurse roster is evaluated using an objective function (see (4)) that adds up the penalty of soft constraint violations in a feasible roster.

$$\min f(\mathbf{x}) = \sum_{s=1}^{10} c_s \cdot g_s(\mathbf{x}). \tag{4}$$

Note that  $s$  refers to the index of the soft constraint ( $S_1, \dots, S_{10}$ ),  $c_s$  refers to the penalty weight for the violation of the soft constraint  $s$ , and  $g_s(\mathbf{x})$  is the total number of violations in  $\mathbf{x}$  for the soft constraint  $s$ , where  $\mathbf{x}$  is a roster solution as formulated in Fig. 1.

### 3 Modified Harmony Search Algorithm for NRP

The HSA is an optimization technique inspired by the music improvisation process. Naturally, musicians play pitches of their instruments relying on both experiences and randomness realized in their off-hand skills. In the optimization process, decision variables can be assigned based on accumulative search or randomness.

The HSA is a population-based method starting with a set of vectors stored in *Harmony Memory* (HM). At each iteration, a new vector (i.e., *new harmony*) is generated based on three operators: (i) *Memory Consideration*, which makes use of accumulative search from HM vectors (i.e., same functionality as the crossover operator in Genetic Algorithm (GA)); (ii) *Random Consideration*, which is used to diversify the new harmony (i.e., same functionality as the mutation operator in GA), and (iii) *Pitch Adjustment*, which is responsible for the local improvement (i.e., same functionality as move in local search). The objective function is used to evaluate the quality of the new harmony. Iteratively, if the new harmony has a better quality than the worst vector in HM, the HSA will substitute the worst vector with the new harmony and this process is repeated until a stopping criterion is met.

This section thoroughly describes the methodology followed in this paper with detailed explanation of how to modify HSA steps into NRP.

**STEP 1. Initialize the parameters of the NRP and HSA.** Within the NRP, the parameters are normally drawn from the dataset instance to be processed. These parameters include the set of nurses, the set of skill categories, the set of shift types, the scheduling period times, the set of work contracts, the set of cover demand requests, the set of preferences of nurses, and eventually the set of unwanted patterns. Each contract contains the details of agreement between the hospital and the nurse that include: maximum/minimum number of assignment shifts, maximum/minimum number of consecutive working days, maximum/minimum number of consecutive free days, maximum/minimum number of consecutive working weekends, maximum number of working weekends within four weeks, the days of weekend, and unwanted patterns (see Table 1).

The objective function described in (4) is utilized to evaluate each roster generated by HSA. Fig. 1 displays the roster  $x$  which includes a set of allocations each of which takes a value of a combination of nurses, days, and shifts. The possible range for each allocation is within all possible combinations of nurses, days, and shifts. The parameters of the HSA required to solve the optimization problem are also specified in this step: (i) The Harmony Memory Consideration Rate (HMCR), determines the rate of selecting the values from HM vectors. (ii) The Harmony Memory Size (HMS), determines the number of initial vectors in HM. (iii) The Pitch Adjustment Rate (PAR), determines the rate of the local improvement. (iv) The Number of Improvisations (NI) corresponds to the number of iterations that are required to solve NRP which are also defined.

$x_1$	$x_2$	$x_3$	$x_4$	...	$x_{N-1}$	$x_N$
Nurse 1	Nurse 9	Nurse 2	Nurse 2	...	Nurse 4	Nurse 10
Day 1	Day 12	Day 3	Day 16	...	Day 10	Day 28
Shift D	Shift L	Shift N	Shift E	...	Shift D	Shift DH

**Fig. 1.** Roster  $x$  representation

**STEP 2. Initialize the harmony memory.** The Harmony Memory (**HM**) consists of a set of all feasible rosters as determined by HMS (see (5)). In this step, these rosters are constructed using a heuristic ordering method [31] whereby the shifts will be sorted in ascending order based on their difficulty level, and then the required nurses of the ordered shifts will be assigned first. These rosters are sorted in ascending order based on their objective function values.

$$HM = \left[ \begin{array}{cccc|c} x_1^1 & x_2^1 & \dots & x_N^1 & f(x^1) \\ x_1^2 & x_2^2 & \dots & x_N^2 & f(x^2) \\ \dots & \dots & \dots & \dots & \dots \\ x_1^{HMS} & x_2^{HMS} & \dots & x_N^{HMS} & f(x^{HMS}) \end{array} \right] \quad (5)$$

**STEP 3. Improvise a new harmony roster.** In this step, a new harmony roster,  $x' = (x_1', x_2', \dots, x_N')$ , is constructed from scratch based on three operators: (i) memory consideration, (ii) random consideration, and (iii) pitch adjustment. The new roster must be complete and feasible. It should be emphasized that in certain iteration, if the HSA cannot construct a complete and feasible roster, the repair procedure will be triggered.

*Memory Consideration operator.* The memory consideration operator selects a feasible value for allocation  $x_i'$  in the new harmony roster from the values of the same allocations stored in HM rosters such that  $x_i' \in \{x_i^1, x_i^2, \dots, x_i^{HMS}\}$ , for  $\forall i \in (1, 2, \dots, N)$  with probability (w.p.) of HMCR where  $HMCR \in [0,1]$ . Notice that the feasibility must be always maintained.

*Random Consideration operator.* The allocation  $x_i'$  that met the probability of  $(1 - HMCR)$  will be assigned by a random value from its possible range where the rules of heuristic ordering method have been considered. The process of Memory Consideration and Random Consideration can be summarized as follows:

$$x_i' = \begin{cases} x_i' \in \{x_i^1, x_i^2, \dots, x_i^{HMS}\} & \text{w.p. } HMCR, \\ x_i' \in X_i' & \text{w.p. } (1 - HMCR). \end{cases}$$

Where  $X_i'$  is a set of all feasible values for allocation  $x_i'$ .

*Pitch Adjustment operator.* The allocation  $x_i'$  assigned by memory consideration will be pitch adjusted with probability of PAR where  $PAR \in [0,1]$  as follows:

$$\text{Pitch adjustment for } x_i' = \begin{cases} Yes & \text{w.p. } PAR, \\ No & \text{w.p. } (1 - PAR). \end{cases}$$

For NRP, different neighbourhood structures have been used to improve the roster locally. Here, the pitch adjustment operator is divided into four local changes: (i) **Move**, (ii) **Swap1**, (iii) **Swap2**, and (iv) **Switch**. Each of which is controlled by a specific PAR range as follows:

$$x_i' = \begin{cases} \text{Move} & 0 \leq U(0, 1) < (\text{PAR}/4), \\ \text{Swap1} & (\text{PAR}/4) \leq U(0, 1) < (2 \times \text{PAR}/4), \\ \text{Swap2} & (2 \times \text{PAR}/4) \leq U(0, 1) < (3 \times \text{PAR}/4), \\ \text{Switch} & (3 \times \text{PAR}/4) \leq U(0, 1) < \text{PAR}, \\ \text{Do nothing} & \text{PAR} \leq U(0, 1) \leq 1. \end{cases}$$

The four proposed pitch adjustment procedures are designed to run as follows:

1. **Move:** with probability of  $[0, \text{PAR}/4)$ , the nurse of the selected allocation  $x_i'$  will be changed to another nurse randomly to solve the violation of the soft constraint S4 (see Fig.2. (a)).
2. **Swap1:** with probability of  $[\text{PAR}/4, 2 \times \text{PAR}/4)$ , the shift of the selected allocation  $x_i'$  will be exchanged with another shift on the same day for another selected allocation  $x_j'$  to solve the violation of the soft constraint S10 (see Fig.2. (b)).
3. **Swap2:** with probability of  $[2 \times \text{PAR}/4, 3 \times \text{PAR}/4)$ , the shift of the selected allocation  $x_i'$  will be exchanged with another shift on the same day for another selected allocation  $x_j'$  to solve or minimize the violations of the soft constraints S5, S8, S10 (see Fig.2. (c)).
4. **Switch:** with probability of  $[3 \times \text{PAR}/4, \text{PAR})$ , the shift of the selected allocation  $x_i'$  will be exchanged with another shift with the same nurse for another selected allocation  $x_j'$  to solve or minimize the violations of the soft constraints S5, S8, S10 (see Fig.2.(d)).

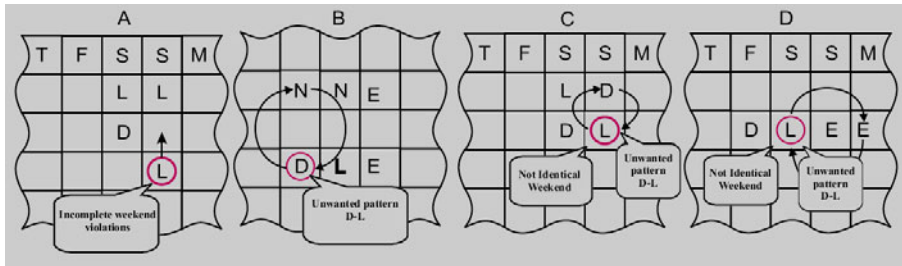


Fig. 2. Pitch Adjustment procedures

In this paper, any local changes that do not improve the new harmony or result in an unfeasible roster will be discarded. Note that the other constraints not covered in the pitch adjustment procedures; do not apply to a partial roster, but rather to a complete one.

**STEP 4. Repair the new harmony.** When the improvisation (STEP 3) process of the new harmony roster is completed, it must be checked for feasibility. However, if the feasibility is not achieved, then the repair process is triggered to render the new harmony roster feasible using the following two steps: Firstly, identify the allocations which are not scheduled in the new harmony roster. Secondly, use a random selection to find feasible values for unscheduled allocations in the new harmony roster. Yet, if the repair process with the predefined iterations is not successful in solving the feasibility issue, then discard the new harmony roster and restart the improvisation process from scratch.

**STEP 5. Update the harmony memory.** If the new roster,  $\mathbf{x}' = (x_1', x_2', \dots, x_N')$ , is better than the worst roster in **HM**, the new roster replaces the worst roster.

**STEP 6. Check the stop criterion.** Based on NI step 3, step 4, and Step 5 of HSA are repeated.

## 4 Experiments and Results

The MHSA evaluated in this section is programmed using Microsoft Visual C++ 6.0 under WinXP on an Intel Machine with CoreTM processor 2.66GHz, and 2GB RAM. A dataset provided by INRC2010 for nurse rostering is used. The Sprint group of INRC2010 dataset used here consists of 33 problem instances (10 sprint early, 10 sprint late, 10 sprint hidden, 3 sprint hint). The characteristics of the dataset are overviewed in Table 3. The conducted experiments are designed to investigate the effectiveness of MHSA for NRP. Each dataset ran 10 times. The parameter settings are: HMS =100, HMCR=0.99, PAR=0.001 and NI is between 100000 and 300000.

**Table 3.** Characteristics of the sprint group of INRC2010 dataset

<b>Characteristic</b>	<b>Early</b>	<b>Late</b>	<b>Hidden</b>	<b>Hint</b>
<i>Number of nurses</i>	10	10	10	10
<i>Number of skills</i>	1	1	1	1
<i>Number of shifts</i>	4	4	3, 4	4
<i>Number of contracts</i>	4	3	3	3
<i>Number of patterns</i>	3	0, 3, 4, 8	4, 8	0, 8
<i>Period of schedule</i>	1 to 28 Jan. 2010	1 to 28 Jan. 2010	1 to 28 Jun. 2010	1 to 28 Jan. 2010
<i>Day-Off request</i>	√	√, or X	√	√
<i>Shift-Off request</i>	√	√, or X	√	√

Table 4 shows the results produced by MSHA in terms of best, average, worst and standard deviation. Moreover, the competition results are recorded for the purpose of comparison. Note that the numbers in the table refer to the penalty values of the violations on the soft constraints (lowest is best). The best results are highlighted in bold while the underlined results are the close-to-the-best results achieved by MHSA. The values of differences between the best cited result and MHSA results are also shown in the *Diff.* column. The symbol '√' indicates that the used method obtains the best result while the symbol '-' denotes that it cannot obtain the best result.

Apparently, the results are comparable and impressive, yet cannot precisely measure up to the best cited results. However, these results can be excused as this method is experimented with to initially explore its efficiency for NRP.

The results are compared with competition participants' methods abbreviated as follows<sup>2</sup>

- G1: A hyper-heuristic combined with a greedy shuffle approach by Burak Bilgin, Peter Demeester, Mustafa Misir, Wim Vancroonenburg, Greet Vanden Berghe, and Tony Wauters.

<sup>2</sup><http://www.kuleuven-kortrijk.be/.u00411139/nrpcompetition/abstracts/>



- G2: An ejection chain method and a branch and price algorithm by Edmund K. Burke, and Tim Curtois.
- G3: Adaptive Local Search, by Zhipeng Lu and Jin-Kao Hao.
- G4: General Constraint Optimization Solver by Koji Nonobe.
- G5: A systematic two-phase approach by Christos Valouxis, Christos Gogos, George Goulas, Panayiotis Alefragis and Efthymios Housos.

**Table 4.** The results achieved by MHSA for NRP

Instance Name	MHSA Results				Best Results	Diff.	Competition Winners				
	Best	Worst	Average	Std. div.			G1	G2	G3	G4	G5
Sprint Early 01	60	73	65.1	3.59	56	4	-	√	√	√	√
Sprint Early 02	61	76	66.4	4.22	58	3	-	√	√	√	√
Sprint Early 03	56	65	59.9	2.98	51	5	-	√	√	√	√
Sprint Early 04	66	81	71.9	4.46	59	7	√	√	-	√	√
Sprint Early 05	61	69	65	2.61	58	3	√	√	√	√	√
Sprint Early 06	58	69	63.6	3.29	54	4	√	√	√	√	√
Sprint Early 07	62	81	65.8	5.38	56	6	√	√	√	√	√
Sprint Early 08	59	65	62.8	2.04	56	3	-	√	√	√	√
Sprint Early 09	57	69	63.6	3.47	55	2	-	√	√	√	√
Sprint Early 10	58	70	62	3.52	52	6	-	√	√	√	√
Sprint Late 01	47	58	53.8	3.43	37	10	-	√	-	-	√
Sprint Late 02	53	65	58.1	3.08	42	11	-	√	-	-	√
Sprint Late 03	59	71	65.1	4.04	48	11	-	√	-	√	√
Sprint Late 04	117	138	127.6	6.33	75	42	-	√	-	-	-
Sprint Late 05	54	63	57.5	3.07	44	10	-	√	-	-	√
Sprint Late 06	47	68	54.6	6.39	42	5	-	√	√	√	√
Sprint Late 07	66	107	87.3	11.40	42	24	-	√	-	-	-
Sprint Late 08	19	81	45.1	17.28	17	2	-	√	√	√	√
Sprint Late 09	34	99	55.5	18.33	17	17	-	√	√	√	√
Sprint Late 10	73	116	95.7	13.39	43	30	-	√	-	-	-
Sprint Hidden 01	48	74	54.8	6.76	33	15	-	-	-	√	√
Sprint Hidden 02	45	61	52.1	4.64	32	13	√	-	-	√	-
Sprint Hidden 03	76	89	81.4	3.61	62	14	-	-	-	√	√
Sprint Hidden 04	97	214	178.4	33.99	67	30	-	-	-	√	√
Sprint Hidden 05	68	89	77.5	7.10	59	9	√	-	-	-	-
Sprint Hidden 06	278	977	481.9	199.68	134	144	-	-	-	√	-
Sprint Hidden 07	201	374	292.4	48.71	153	48	-	-	-	-	√
Sprint Hidden 08	374	488	432	31.94	209	165	-	-	-	√	-
Sprint Hidden 09	916	1357	1232.8	125.50	338	578	-	-	-	-	√
Sprint Hidden 10	462	605	542	42.13	306	156	-	-	-	-	√
Sprint Hint 01	104	133	118.9	8.68	78	26	-	√	-	-	-
Sprint Hint 02	73	98	87.7	7.25	47	26	-	√	-	-	-
Sprint Hint 03	92	131	112.8	10.68	57	35	-	√	-	-	-

## 5 Conclusion and Future Work

This paper has presented a Modified Harmony Search Algorithm (MHSA) for the Nurse Rostering Problem (NRP) as an initial investigation to experiment with the method for tackling NRP. Firstly, heuristic ordering has been used to generate feasible solutions that satisfy all hard constraints. Secondly, harmony search operators are adapted to realize the problem domain knowledge. Note that the feasible search space region is dealt with. MHSA is able to generate a new roster in each iteration;

the roster is globally improved using the memory consideration, random consideration, and is locally improved using pitch adjustment. Using the International Nurse Rostering Competition 2010 (INRC2010) dataset, the MHSA is able to produce a feasible roster with competitively comparable results.

In the future, a sensitivity analysis for the HSA parameters for NRP has to be carried out, and more work could be done on developing new neighbourhood techniques based on different problem constraints or hybridizing MHSA with local search-based methods. We believe that powerful local changes will substantially improve the quality of the results.

**Acknowledgments.** The third author is grateful to be awarded a Postdoctoral Fellowship from the school of Computer Sciences (USM).

## References

1. Aickelin, U., Dowsland, K.: An indirect genetic algorithm for a nurse-scheduling problem. *Computers & Operations Research* 31(5), 76–778 (2004)
2. Al-Betar, M., Khader, A.: A hybrid harmony search for university course timetabling. In: *Proceedings of the 4nd Multidisciplinary Conference on Scheduling: Theory and Applications (MISTA 2009)*, Dublin, Ireland, pp. 10–12 (August 2009)
3. Al-Betar, M., Khader, A.: A harmony search algorithm for university course timetabling. *Annals of Operations Research*, 1–29 (2008)
4. Al-Betar, M., Khader, A., Nadi, F.: Selection mechanisms in memory consideration for examination timetabling with harmony search. In: *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, pp. 1203–1210. ACM (2010)
5. Alatas, B.: Chaotic harmony search algorithms. *Applied Mathematics and Computation* 216(9), 2687–2699 (2010)
6. Bilgin, B., De Causmaecker, P., Rossie, B., Vanden Berghe, G.: Local search neighbourhoods for dealing with a novel nurse rostering model. *Annals of Operations Research*, 1–25 (2011)
7. Brusco, M., Jacobs, L.: Cost analysis of alternative formulations for personnel scheduling in continuously operating organizations. *European Journal of Operational Research* 86(2), 249–261 (1995)
8. Burke, E., Cowling, P., De Causmaecker, P., Berghe, G.: A memetic approach to the nurse rostering problem. *Applied Intelligence* 15(3), 199–214 (2001)
9. Burke, E., Curtois, T., Post, G., Qu, R., Veltman, B.: A hybrid heuristic ordering and variable neighbourhood search for the nurse rostering problem. *European Journal of Operational Research* 188(2), 330–341 (2008)
10. Burke, E., Curtois, T., Qu, R., Berghe, G.: A scatter search methodology for the nurse rostering problem. *Journal of the Operational Research Society* 61(11), 1667–1679 (2009)
11. Burke, E., De Causmaecker, P., Berghe, G., Van Landeghem, H.: The state of the art of nurse rostering. *Journal of Scheduling* 7(6), 441–499 (2004)
12. Burke, E., De Causmaecker, P., Vanden Berghe, G.: A hybrid tabu search algorithm for the nurse rostering problem. *Simulated Evolution and Learning*, 187–194 (1999)
13. Burke, E., Li, J., Qu, R.: A hybrid model of integer programming and variable neighbourhood search for highly-constrained nurse rostering problems. *European Journal of Operational Research* 203(2), 484–493 (2010)

14. Cheang, B., Li, H., Lim, A., Rodrigues, B.: Nurse rostering problems - a bibliographic survey. *European Journal of Operational Research* 151(3), 447–460 (2003)
15. Geem, Z.: Optimal cost design of water distribution networks using harmony search. *Engineering Optimization* 38(3), 259–277 (2006)
16. Geem, Z.: Harmony search applications in industry. *Soft Computing Applications in Industry*, 117–134 (2008)
17. Geem, Z., Kim, J., Loganathan, G.: A new heuristic optimization algorithm: harmony search. *Simulation* 76(2), 60–68 (2001)
18. Geem, Z., Lee, K., Park, Y.: Application of harmony search to vehicle routing. *American Journal of Applied Sciences* 2(12), 1552–1557 (2005)
19. Geem, Z., Sim, K.: Parameter-setting-free harmony search algorithm. *Applied Mathematics and Computation*, 3881–3889 (2010)
20. Geem, Z., Tseng, C., Park, Y.: Harmony search for generalized orienteering problem: best touring in china. *Advances in Natural Computation*, 741–750 (2005)
21. Gutjahr, W., Rauner, M.: An ACO algorithm for a dynamic regional nurse-scheduling problem in Austria. *Computers & Operations Research* 34(3), 642–666 (2007)
22. Kaveh, A., Talatahari, S.: Particle swarm optimizer, ant colony strategy and harmony search scheme hybridized for optimization of truss structures. *Computers & Structures* 87(5-6), 267–283 (2009)
23. Lee, K., Geem, Z.: A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. *Computer Methods in Applied Mechanics and Engineering* 194(36-38), 3902–3933 (2005)
24. Lee, K., Geem, Z., Lee, S., Bae, K.: The harmony search heuristic algorithm for discrete structural optimization. *Engineering Optimization* 37(7), 663–684 (2005)
25. Maenhout, B., Vanhoucke, M.: An electromagnetic meta-heuristic for the nurse scheduling problem. *Journal of Heuristics* 13(4), 359–385 (2007)
26. Mahdavi, M., Fesanghary, M., Damangir, E.: An improved harmony search algorithm for solving optimization problems. *Applied Mathematics and Computation* 188(2), 1567–1579 (2007)
27. Millar, H., Kiragu, M.: Cyclic and non-cyclic scheduling of 12 h shift nurses by network programming. *European Journal of Operational Research* 104(3), 582–592 (1998)
28. Pan, Q., Suganthan, P., Liang, J., Tasgetiren, M.: A local-best harmony search algorithm with dynamic subpopulations. *Engineering Optimization* 42(2), 101–117 (2010)
29. Pan, Q., Suganthan, P., Tasgetiren, M., Liang, J.: A self-adaptive global best harmony search algorithm for continuous optimization problems. *Applied Mathematics and Computation* 216(3), 830–848 (2010)
30. Wren, A.: Scheduling, Timetabling and Rostering a Special Relationship? In: Burke, E.K., Ross, P. (eds.) *PATAT 1995*. LNCS, vol. 1153, pp. 46–75. Springer, Heidelberg (1996)
31. Zhao, S., Suganthan, P., Pan, Q., Tasgetiren, M.: Dynamic multi-swarm particle swarm optimizer with harmony search. *Expert Systems with Applications*, 3735–3742 (2010)

# A Swarm Intelligence Based Algorithm for QoS Multicast Routing Problem

Manoj Kumar Patel, Manas Ranjan Kabat, and Chita Ranjan Tripathy

Department of Computer Science and Engineering,  
Veer Surendra Sai University of Technology, Burla, India  
patel.mkp@gmail.com, {manas\_kabat, crt.vssut}@yahoo.com

**Abstract.** The QoS multicast routing problem is to find a multicast routing tree with minimal cost that can satisfy constraints such as bandwidth, delay, delay jitter and loss rate. This problem is NP Complete. In this paper, we present a swarming agent based intelligence algorithm using a hybrid Ant Colony Optimization/Particle Swarm Optimization (ACO/PSO) algorithm to optimize the multicast tree. The algorithm starts with generating a large amount of mobile agents in the search space. The ACO algorithm guides agents' movement by pheromones in the shared environment locally and the global maximum of the attribute values are obtained through the random interaction between the agents using PSO algorithm. The performance of the proposed algorithm is evaluated through simulation. The simulation results reveal that our algorithm performs better than the existing algorithms.

## 1 Introduction

The rapid development in network multimedia technology enables more and more real-time multimedia services such as video conferencing, on-line games, distance education etc. to become mainstream Internet activities. These services often require the underlying network to provide multicast capabilities. Multicast refers to the delivery of packets from a single source to multiple destinations. These real-time applications have a stringent requirement of QoS parameters like bandwidth, delay, jitter and so on to ensure smooth, consistent and fair transmission to the receivers. The central problem of QoS routing is to set up a multicast tree that can satisfy certain QoS parameters. However, the problem of constructing a multicast tree under multiple constraints is NP-Complete [1]. Hence, the problem is usually solved by heuristic or intelligence optimization.

In recent years, many researchers have adopted meta-heuristic algorithm such as ant colony algorithm [2-8] and particle swarm optimization [9-13] to solve multi-constrained QoS routing problems. An intelligent routing algorithm ANTNET based on ant colony algorithm was proposed in [2, 3]. Similarly, other ant intelligence algorithm was introduced in [5] for the computation of the QoS multicast tree. A tree growth based ACO algorithm has been proposed in [8] to generate a multicast tree in the way of tree growth and optimizing ant colony parameters through the most efficient combination of various parameters. The general weakness of ant colony

algorithm is that it converges slowly at the initial step and takes more time to converge which is due to improper selection of initial feasible parameter [4]. The overhead also increases due to merging and pruning of trees.

Many scholars have discussed the application of Particle Swarm Optimization (PSO) algorithm to solve QoS constraint routing problem. The PSO algorithm proposed in [12] to solve the QoS multicast routing problem can obtain a feasible multicast tree by exchanging the paths. This algorithm can converge to the optimal or near optimal solution with lower computational cost. Besides this other algorithm based on quantum mechanics named as Quantum-Behaved Particle Swarm Optimization (QPSO) was proposed [9]. Later on PSO along with Genetic Algorithm (GA) was introduced which become hybrid genetic algorithm and particle swarm Optimization (HGAPSO) [10] to solve multicast QoS routing. A tree based PSO has been proposed in [13] for optimizing the multicast tree directly. However, the performance depends on the number of particles generated. Another drawback of the algorithm is merging the multicast trees, eliminating directed circles and nested directed circles are very complex.

In this paper, we propose a hybrid ACO/PSO algorithm based on the swarming agent architecture for QoS multicast routing. Our work is inspired by the swarming agent algorithm proposed by Brueckner and Parunak [14] for distributed data pattern and Y. Meng [15] for Proteomic Pattern detection of ovarian cancer. Basically, large amount of mobile agents are generated in the search space. Two collective and co-ordination process for the mobile agents are proposed. One is based on the ACO [8] algorithm for guiding the agents' movements by pheromones in the shared environment locally and the PSO algorithm [13] for obtaining the global maximum of the attribute values through the random interaction between the agents.

The rest of the paper is organized as follows. The mathematical model developed to model a computer network is presented in section 2. The proposed algorithm and its principle of working are presented in section 3. The simulation results are presented in section 4. Finally, the concluding remarks are presented in Section 5.

## 2 Problem Statement

A network is modeled as a directed, connected graph  $G = (V, E)$ , where  $V$  is a finite set of vertices (network nodes) and  $E$  is the set of edges (network links) representing connection of these vertices. Let  $n = |V|$  be the number of network nodes and  $l = |E|$  be the number of network links. The link  $e = (u, v)$  from node  $u$  to  $V$  to node  $v$  to  $V$  implies the existence of a link  $e' = (v, u)$  from node  $v$  to node  $u$ . Four non-negative real value functions are associated with each link  $e(e \in E)$ : cost  $C(e): E \rightarrow \mathcal{R}^+$ , delay  $D(e): E \rightarrow \mathcal{R}^+$ , loss rate  $L(e): E \rightarrow \mathcal{R}^+$ , and available bandwidth  $B(e): E \rightarrow \mathcal{R}^+$ . Because of the asymmetric nature of the communication networks, it is often the case that  $C(e) \neq C(e')$ ,  $D(e) \neq D(e')$ ,  $L(e) \neq L(e')$ ,  $B(e) \neq B(e')$ .

A multicast tree  $T(s, M)$  is a sub-graph of  $G$  spanning the source node  $s \in V$  and the set of destination nodes  $M \subseteq V - \{s\}$ . Let  $m = |M|$  be the number of multicast destination nodes. We refer to  $M$  as the destination group and  $\{\{s\} \cup M\}$  the multicast group. In addition,  $T(s, M)$  may contain relay nodes (Steiner nodes), the nodes in the

multicast tree but not in the multicast group. Let  $P_T(s, d)$  be a unique path in the tree  $T$  from the source node  $s$  to a destination node  $d \in M$ . The quality of the tree is characterized by the following parameters.

The total cost of the tree  $T(s, M)$  is defined as sum of the cost of all links in that tree and can be given by

$$C(T(s, M)) = \sum_{e \in T(s, M)} C(e)$$

The total delay of the path  $P_T(s, d)$  is simply the sum of the delay of all links along  $P_T(s, d)$ :

$$D(P_T(s, d)) = \sum_{e \in T(P_T(s, d))} D(e)$$

The total loss rate of the path:

$$L(P_T(s, d)) = 1 - \prod_{e \in T(P_T(s, d))} (1 - L(e))$$

The bottleneck bandwidth of the path  $P_T(s, d)$  is defined as minimum available residual bandwidth at any link along the path:

$$B(P_T(s, d)) = \min \{B(e), e \in P_T(s, d)\}$$

The delay jitter of the tree  $T(s, M)$  is defined as the average difference of delay on the path from the source to the destination node:

$$DJ(T(s, M)) = \sqrt{\sum_{d \in M} (D(P_T(s, d)) - \text{delay\_avg})^2}$$

Where  $\text{delay\_avg}$  denotes the average value of delay on the path from the source to the destination nodes.

Let  $\Delta$  be the delay constraint,  $\zeta$  be the loss rate constraint,  $\beta$  be the bandwidth constraint of the path from source to the destination node  $d$ ,  $\delta$  be the delay jitter constraint. The multi-constrained least-cost multicast problem is defined as:

Minimize  $C(T(s, M))$ , subject to :

$$\begin{aligned} D(P_T(s, d)) &\leq \Delta && \forall d \in M \\ L(P_T(s, d)) &\leq \zeta && \forall d \in M \\ B(P_T(s, d)) &\geq \beta && \forall d \in M \\ DJ(T(s, M)) &\leq \delta \end{aligned}$$

### 3 A Hybrid ACO/PSO Algorithm Using Swarming Agents for Multicast Routing

An agent is an independent processing entity that interacts with the external environment and the other agents to pursue its particular set of goals. By using the ACO algorithm, the agents in the systems coordinate their behaviors and communicate their results through pheromone. On the other hand, agents using PSO algorithm coordinates their behaviors through the random interaction with other

agents. Therefore, a hybrid ACO/PSO algorithm is proposed in this paper for the swarming-agent based multicast routing problem.

### 3.1 Swarming Agents Architecture

Initially,  $n$  multicast tree patterns are generated randomly and  $m$  key values as attributes are calculated for  $m$  destinations of each multicast tree pattern. Therefore, the structure of the pattern is defined as the following equation:

$$T_i = \{a_{i1}, a_{i2}, \dots, a_{im}\} \text{ for } i=1..n$$

where  $T_i$  denotes the multicast tree pattern  $i$ ,  $a_{ij}$  represents the  $j^{\text{th}}$  attribute of pattern  $i$ , in each pattern and  $n$  is the number of the generated patterns.

For each pattern an associated *pattern agent* is created which is fixed to each pattern. Then  $n$  mobile agents are generated to detect the fit patterns from the randomly generated pattern and recombine some of the selected patterns together to build a pattern with more fitness value. These mobile agents are referred as *particle agents* who can move from one pattern to another in the search space and interact with other particle agents dynamically. Initially, the particle agents are uniformly distributed in the search space. After many iterations of the algorithm, eventually the particle agents will converge to the fittest pattern.

The particle agents are allowed to deposit pheromones and sense local attributes in each pattern. There are two levels of pheromones, one is for pattern pheromone and the other is attribute pheromone inside the pattern. The pattern with high pheromone has either high probability to become the best fit pattern or some of the attributes inside this pattern has the higher potential to be included in the best fit pattern. This attracts more particle agents to move to the pattern. Once an agent enters into a pattern it deposits pheromone on the selected attributes and deposits the pheromones on the pattern, both are proportional to their fitness values.

### 3.2 Hybrid ACO/PSO Algorithm

The multicast tree patterns are generated randomly and these patterns reside in a search space which is defined as a  $k \times l$  rectangular grid. Given source node  $s$ , the group member  $d_1, d_2, d_3, \dots, d_m$  where  $m$  is the number of group members. The pseudo code for multicast tree generation is given below.

```

procedure PatternGeneration( $T_i$ )
1. Begin
2. initialize  $V_n = \{s\}$ 
3. delaysofar( $s$ )=0, losssofar( $s$ )=0, costsofar( $s$ )=0;
4. cur_node = s
5. repeat
6.    $N_{\text{cur\_node}} = \phi$ 
7.   for each neighbour node  $v$  of the cur_node
           such that  $v \notin V_n$ 
8.     if( $B(\text{cur\_node}, v) \geq \beta$  and delaysofar(cur_node) +

```

```

    D(cur_node) ≤ Δ and 1 - (losssofar(cur_node) *
    (1 - L(cur_node, v)) ≤ ζ)
9.     Ncur_node = Ncur_node ∪ {v}
10.    end if
11.  end for
12.  j = SelectRandom(Ncur_node)
13.  cost = cost + C(cur_node, j)
14.  costsofar(j) = costsofar(cur_node) + C(cur_node, j)
15.  pheromone(j) = 1/costsofar(j)
16.  delaysofar(j) = delaysofar(cur_node) + D(cur_node, j)
17.  losssofar(j) = losssofar(cur_node) * (1 - L(cur_node, j))
18.  if( j ∉ M ){
19.    cur_node = j
20.  else
21.    cur_node = SelectRandom(VT)
22.    for all( u ∈ M and u ∈ VT)
23.      pheromone(u) = pheromone(u) + pheromone(cur_node)
24.    end for all
25.  end if
26. until (VT contains all nodes of the multicast group)
27. if Ti satisfies the delay jitter that is DJ(Ti(s, M)) ≤ δ
28. then calculate pheromone(Ti) from the cost of the tree
29. return Ti
30. end PatternGeneration

```

Initially, the multicast tree  $V_T$  is set to the source node  $s$ , which is also set as the current node. Accumulated delay and loss up to source node  $s$ ,  $delaysofar(s)$  and  $losssofar(s)$  are set to 0 and 1 respectively. Then, a node  $j$  is selected randomly from the neighbor nodes of the current node that satisfied the QoS constraints. The node  $j$  is added to the multicast  $V_T$  and the  $delaysofar(j)$  and  $losssofar(j)$  upto node  $j$  is calculated as shown in step 14 and 15 respectively. If the node  $j$  is not the destination node then the current node is set to  $j$  otherwise the current node is selected randomly from  $V_T$ . The pheromones of the multicast group member that are already added to the multicast tree are updated as shown in step 23. This process is repeated till all the group members are added to the multicast tree.

These tree patterns are filled into the grids to the search space, where each pattern is corresponding into one grid, based on the order of their generation. Then,  $n$  particle agents are generated and uniformly distributed to the search space, where each particle agent occupies one grid. For each iteration, the particle agent evaluates the fitness of the tree pattern from the cost of the tree pattern. Due to the fixed topology for the patterns in the search grid, the particle agent can interact with the maximum 8 neighbor particle agents. If the fitness of the best neighbor is more than the current pattern then the local based pattern is combined with the current pattern to generate a new pattern.

The particle moves to the new position and brings the last position's attribute and its associated fitness values along with its movement to the new position. The new



position is updated with better quality attributes discarding the low quality attribute of the new position. The outcome is a new combined pattern with higher quality than both original ones. During the next iteration, the newly built pattern will be executed by the agents and deposit the pheromone as appropriate. After much iteration, eventually the most strong pheromone trail will be the fittest discriminating pattern.

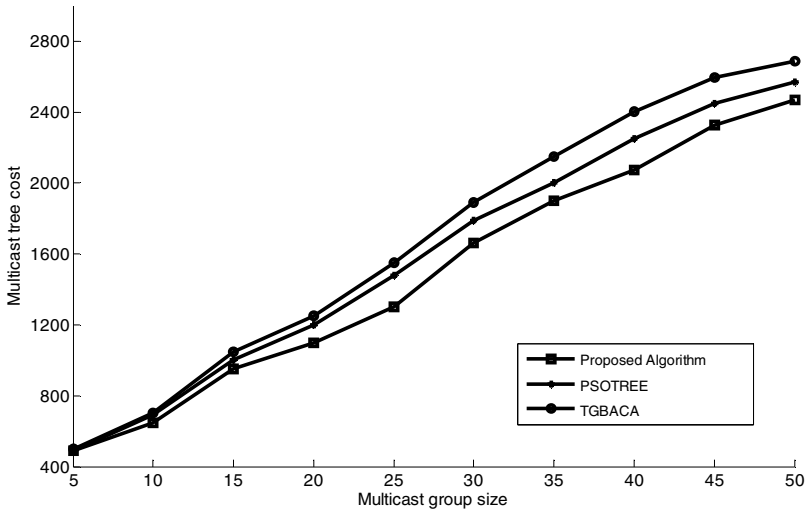


Fig. 1. Multicast tree cost vs. group size (No of nodes=100)

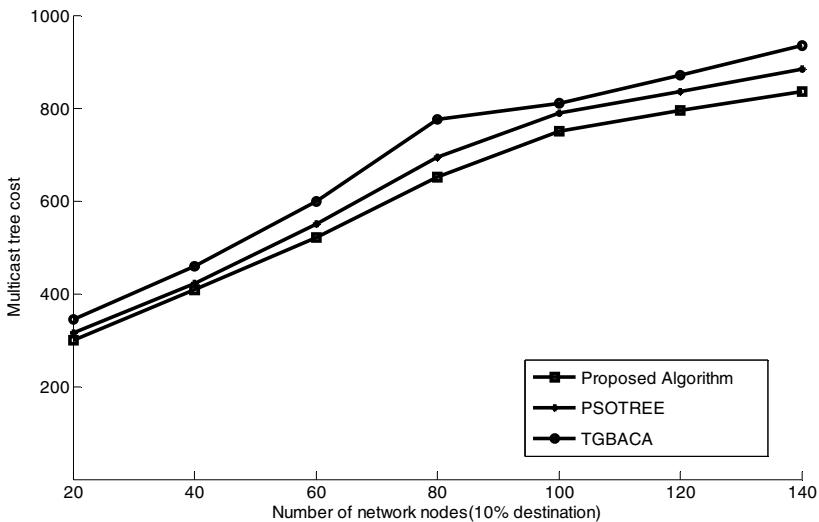


Fig. 2. Multicast tree cost vs. Network size with 10% nodes as destinations

## 4 Simulation Results

We have implemented our proposed algorithm in Visual C<sup>++</sup>. The experiments are performed on an Intel Core i3 @ 2.27 G.Hz. and 2 GB RAM based platform running Windows 7.0.

The positions of the nodes are fixed randomly in a rectangle of size 4000 km x 2400 km. The Euclidean metric is then used to determine the distance between each pair of nodes. The network topology used in our simulation was generated randomly using Waxman's topology [16]. Edges are introduced between the pairs of nodes  $u, v$  with a probability that depends on the distance between them. The edge probability is given by  $P(u, v) = \beta \exp(-l(u, v) / \alpha L)$ , where  $l(u, v)$  is the Euler distance from node  $u$  to  $v$  and  $L$  is the maximum distance between any two points in the network. The delay, loss rate and band width of the links are set randomly from 1 to 30, 0.0001 to 0.01 and 2 to 10 Mbps respectively.

The source node is selected randomly and destination nodes are picked up uniformly from the set of nodes chosen in the network topology. The delay bound  $\Delta$ , the delay jitter bound and the loss bound are set 120ms, 60ms and 0.05 respectively. The bandwidth requested by a multicast application is generated randomly. We also implement PSOTREE [13] and TGBACA [8] algorithms in the same environment to study and compare the performance of our proposed algorithm with the existing algorithms. We generate 30 multicast trees randomly to study the performance of PSOTREE and TGBACA and 30 multicast trees for our proposed algorithm. These 30 multicast tree patterns are arranged in a rectangular grid of size 5x6 in the order of their generation. The simulation is run for 100 times for each case and the average of the multicast tree cost is taken as the output.

The multicast tree cost verses multicast group size for a network of 100 nodes is shown in Fig. 1. The Fig. 2 shows the multicast tree cost verses the network size with 10 percent of the nodes as the group size. The multicast trees generated by PSOTREE, TGBACA and our proposed algorithm satisfy the delay, delay jitter, loss rate and bandwidth constraints. However, the figures clearly illustrate that the cost of the multicast tree generated by our proposed algorithm is less than the multicast trees generated by PSOTREE and TGBACA. The PSOTREE algorithm constructs the multicast tree by combining the multicast trees and removing directed cycles. This algorithm removes the links that are in any of the trees, but not in both and have minimum fitness. However, this approach may not generate a better tree, because the links deleted from the cycle may be better than the links not in the directed cycles. The TGBACA algorithm follows a pheromone updating strategy to construct the best multicast tree. The algorithm updates pheromones on the links used by the global best tree and the best tree generated after each generation. Though this strategy fasts the convergence process, but the solution may fall into local optimization.

In comparison to PSOTREE and TGBACA, our algorithm combines two multicast tree patterns by bringing the better attributes of one pattern to another pattern. Our algorithm generates a new tree pattern after each iteration, which is better than both the patterns. Therefore, our algorithm converges to a multicast tree after a few iterations, which is better than the multicast trees generated by both PSOTREE and TGBACA.

## 5 Conclusions

This paper presents a swarming agent based intelligence algorithm using a hybrid Ant Colony Optimization/Particle Swarm Optimization (ACO/PSO) algorithm for QoS multicast routing. The proposed algorithm generates an economic multicast tree that satisfies delay, delay jitter, bandwidth and loss rate constraints. Our algorithm constructs the multicast tree by combining a few fit multicast tree patterns out of a set of randomly generated multicast tree patterns. In comparison to the existing algorithms [8, 13], we create more patterns and select some fit patterns for the multicast tree construction. Therefore, our algorithm performs better than the existing algorithms [8, 13].

## References

1. Wang, Z., Crowcroft, J.: Quality of service for supporting multimedia application. *IEEE Journal on Selected Areas in Communication* 14, 1228–1234 (1996)
2. Di Caro, G., Dorigo, M.: AntNet: a mobile agents for adaptive routing. In: *Proceedings of the 31st Hawaii International Conference on Systems*, pp. 74–83 (1998)
3. Di Caro, G., Dorigo, M.: AntNet: distributed stigmergetic control for communications networks. *Journal of Artificial Intelligence Research* 9, 317–365 (1998)
4. Dorigo, M., Di Caro, G.: *The ant colony optimization meta-heuristic, new ideas in optimization*. McGraw-Hill (1999)
5. Sim, K.M., Sun, W.H.: Ant colony optimization for routing and load balancing: survey and new directions. *IEEE Transactions on Systems, Man, and Cybernetics* 33, 560–572 (2003)
6. Cheng, H., Cao, J., Wang, X.: A heuristic multicast algorithm to support QoS group communications in heterogeneous network. *IEEE Transactions on Vehicular Technology* 55(3), 831–838 (2006)
7. Mullen, R., Monekosso, D., Barman, S., Remagnino, P.: A review of ant algorithms. *Expert Systems with Applications* 36(6), 9608–9617 (2009)
8. Wang, H., Xu, H., Yi, S., Shi, Z.: A tree-growth based ant colony algorithm for QoS multicast routing problem. *Expert Systems with Applications* 38, 11787–11795 (2011)
9. Sun, J., Liu, J., Xu, W.-b.: QPSO-Based QoS Multicast Routing Algorithm. In: Wang, T.-D., Li, X., Chen, S.-H., Wang, X., Abbass, H.A., Iba, H., Chen, G.-L., Yao, X. (eds.) *SEAL 2006*. LNCS, vol. 4247, pp. 261–268. Springer, Heidelberg (2006)
10. Li, C., Cao, C., Li, Y., Yu, Y.: Hybrid of genetic algorithm and particle swarm optimization for multicast QoS routing. In: *IEEE International Conference on Control and Automation*, pp. 2355–2359 (2007)
11. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: *IEEE International Conference on Neural Network*, Perth, Australia, pp. 1942–1948 (1995)
12. Liu, J., Sun, J., Xu, W.-b.: QoS Multicast Routing Based on Particle Swarm Optimization. In: Corchado, E., Yin, H., Botti, V., Fyfe, C. (eds.) *IDEAL 2006*. LNCS, vol. 4224, pp. 936–943. Springer, Heidelberg (2006)
13. Wang, H., Meng, X., Li, S., Xu, H.: A tree-based particle swarm optimization for multicast routing. *Computer Networks* 54, 2775–2786 (2010)
14. Brueckner, S.A., Parunak, H.V.D.: Swarming agents for distributed pattern detection and classification. In: *AAMAS, Bologna, Italy, July 15-19 (2002)*
15. Meng, Y.: A Swarm Intelligence Based Algorithm for Proteomic Pattern Detection of Ovarian Cancer. In: *IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, Toronto, Canada, September 28–29 (2006)
16. Waxman, B.M.: Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications* 6, 1617–1622 (1988)

# Test Data Generation: A Hybrid Approach Using Cuckoo and Tabu Search

Krish Perumal, Jagan Mohan Ungati, Gaurav Kumar,  
Nitish Jain, Raj Gaurav, and Praveen Ranjan Srivastava

Department of Computer Science and Information Systems, BITS Pilani,  
Vidya Vihar Campus, Pilani – 333031, Rajasthan, India  
{f2008599, f2008387, f2008575, f2008516,  
f2008640, praveenr}@bits-pilani.ac.in

**Abstract.** Software testing is meant to increase confidence in the correctness of software. Due to time, cost and other resource constraints, manual testing is highly impractical and undesirable, especially for the increasingly large sized software being developed these days. Therefore, there is a need to automate the testing process. This calls for the development of a time-efficient technique to automatically generate optimal test data. This paper introduces a novel approach to automated test data generation for software programs using a combination of heuristics involving Cuckoo and Tabu Search. The experimental results have shown a high degree of improvement with respect to the conventional Genetic Algorithm based technique.

## 1 Introduction

Software is one of the most important technologies that will drive the evolution of computer-based systems and products in the current scenario of globalization. In spite of the fact that software has grown from being a mere problem solving and information analysis tool to an industry in itself, a lot is yet to be done on the development of quality software that performs the right job at the right time. It is here that software engineering intends to provide a structured framework for building high quality software [1]. It is with this intention that the Software Development Life Cycle (SDLC) is framed, which is a series of steps that are to be strictly followed in order to produce efficient software in less cost. Amongst the different steps in SDLC, software testing is a very important and mandatory step, which ensures the proper working of the software. Software testing plays an important role in software engineering given the high level of requirement in terms of human resources and time for its efficient completion [1]. Automated test data generation has been one of the most important concerns of software testing [2], particularly owing to the highly limited nature of the prevailing automated tools used for the purpose.

This paper is comprised of 6 sections. The next section deals with the background of the proposed algorithm, explaining the merits and demerits of each of the prevalent methods of automated test data generation and the need for a more holistic meta-heuristic based technique. The 3<sup>rd</sup> section explains the proposed technique, in a step

by step fashion. In the 4<sup>th</sup> section, it is proved that the proposed algorithm is better than the existing approaches through a comparative experimental study. The 5<sup>th</sup> section concludes with the effectiveness of the proposed approach.

## 2 Background

As discussed in the previous section, it is very important to pay attention to the problems involved in automated test data generation. Owing to the same, a qualitative study of the prevalent techniques of test data generation is necessary for a better perspective towards the problems. Hence, this section introduces various schools of thought with regard to the heuristics applied for automating the process of generating test cases/data in general. Their merits and demerits are discussed in the following section.

### *a. Random Testing[3]*

Random testing is the simplest model to be developed for automating the process of generating test data. It does not follow any heuristic rules. It randomly picks up test cases one by one until it finds one that has complete code coverage. Though this model works for all types of test data, it performs with very low efficiency on programs with large codes as well as on those with wide ranges of possible test data.

### *b. Concolic Testing[3]*

Concolic testing is a hybrid technique that makes use of a combination of random and symbolic execution, as explained in [3]. Despite some improvement over random testing, this method is limited by its inability to test nondeterministic programs and its dependence on the symbolic representations used.

### *c. Search Based (Optimization) Testing[3,4,5,6,7]*

The application of search-based or optimization techniques is a very attractive field of study in software testing. It involves the development of a model for test data generation that seeks to optimize a particular objective function, the result of which is complete branch coverage.

The optima of the test data that execute the path could be found using various optimization techniques. A number of optimization techniques like genetic algorithms [4], tabu search [5] and ant colony optimization [6] have yielded favorable results. Experiments [3] on tools using the aforementioned techniques suggest that the concolic and search-based approaches are not as effective for real world programs as researchers may have previously been led to believe by smaller-scale studies. Hence, it is evident that search-based meta-heuristics offer researchers the best chance to optimize the objective functions faster and with greater precision.

It is evident through the above discussion that there are problems of inefficiency and lack of complete code coverage during the application of the aforementioned techniques to specific problems. Therefore, it is proposed to use a new set of heuristics to handle the problem of test data generation in a more holistic manner.

Owing to the perceived improvement in optimization problems by employing meta-heuristic algorithms inspired by nature, a new meta-heuristic cuckoo search is becoming very popular. Experimental results show that cuckoo Search is more generic and robust for many optimization problems, as compared to other meta-heuristic algorithms.

Now, the basics of both cuckoo search and levy flights are explained and how they can be combined and used in order to obtain better results. It is also explained why tabu search is of great importance in improving the memory usage of the proposed algorithm, by preventing the bad solutions from occurring repeatedly.

*Cuckoo Search [8]:* Cuckoo Search in combination with levy flights was developed by Yang and Deb based on the breeding strategy of some cuckoo species.

Cuckoo Search as described by Yang and Deb has its basis in the following three idealized rules:-

- 1) Each cuckoo lays one egg at a time, and dumps its egg in a randomly chosen nest.
- 2) The best nests with high quality of eggs will carry over to the next generations.
- 3) The number of available host nests is fixed, and the egg laid by a cuckoo is discovered by the host bird with a probability  $p_a \in [0, 1]$ . In this case, the host bird can either throw the egg away or abandon the nest, and build a completely new nest.

For simplicity, the last rule can be approximated by the assumption that a fraction  $p_a$  of the  $n$  nests are replaced by new nests (with new random solutions) [8]. When generating new solutions (or cuckoo eggs), a levy flight is preferred, as proposed in [8] for better results. The levy flight essentially provides a random walk while the random step length is drawn from a levy distribution. The random walk via levy flight is more efficient in exploring the search space as its step length is much longer in the long run.

*Tabu Search [9]:* Another technique that is of vital use in optimization problems is the Tabu Search technique. It is a mathematical optimization method that belongs to the class of trajectory based techniques. This technique improves the performance of a local search by marking the previously visited solutions as “tabu”. It ensures that the algorithm does not get stuck in a locally optimum solution, thereby enabling faster convergence to the globally optimal solution.

Despite its many advantages, Tabu Search may perform badly because number of parameters to be determined and the number of resulting iterations could be very large [9]. However, the major effect of these inefficiencies can be reduced to a large extent by choosing the right parameters that may be decided through experimental results.

Considering the above factors, the usefulness of cuckoo search, levy flights [8] and tabu search [9] is well established. Hence, these techniques may be applied to software test data generation to obtain the final solution faster and with better accuracy. The following paragraph discusses how these techniques are modified specifically for use in problems of automated test data generation.

For the generation of new eggs (a test case) and replacing old eggs, cuckoo search is used so as to form a new test case at the end of each iteration. Levy Flight is used in random generation of an egg for ensuring that the new egg is very far from the old one, so as to explore all directions and hence cover all nodes in all different directions. Tabu Search is used to store good test cases as well as bad test cases so that they are not generated redundantly as well as to revert back to a good egg if at all a very bad egg is generated and the solution is stuck at a local optimum. Thus, the employment of all the above techniques ensures less memory wastage and faster convergence to the globally optimal solution.

In the following section, an algorithm is proposed based on the heuristics discussed above that is specifically modified for the purpose of automated test data generation.

### 3 Proposed Technique

In this section, an algorithm is proposed to automate the task of generating test cases for a given program using cuckoo search, tabu search and levy flights. The main motto of the below algorithm is to generate a minimum number of test cases for a given program, such that all the nodes of the program are traversed at least once by these test cases, so as to test for the integrity of the node.

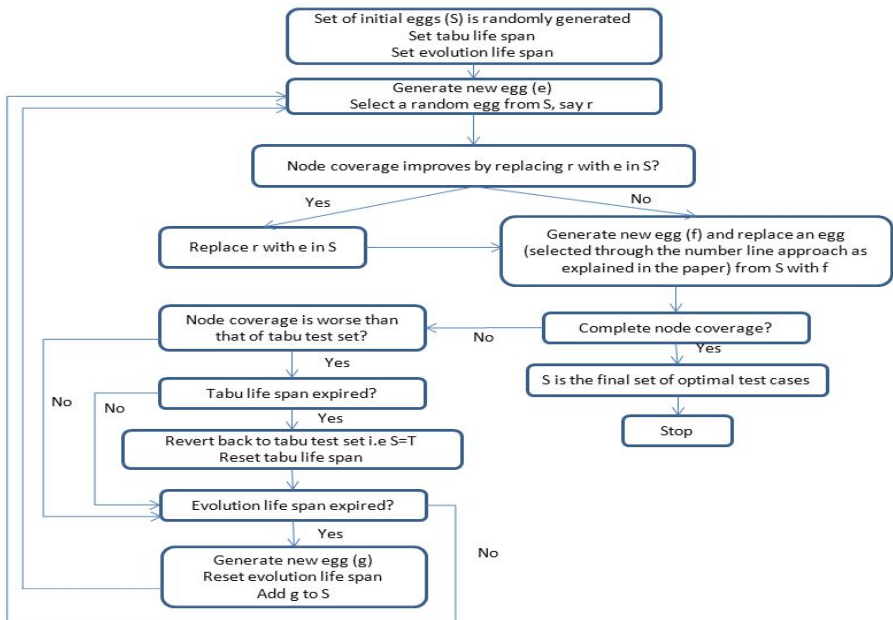


Fig. 1. Flow of Proposed Algorithm

Figure 1 shows the flow of the proposed algorithm. Before discussing the algorithm in more detail, the following terms are introduced.

M = Total number of nodes in CFG

N = Current number of input test cases.

I = Current input test case set.

t = Tabu life-span.

e = Evolution life-span (For which 'N' remains constant).

T = Tabu test case set.

n(S) = Number of nodes covered by a set of test cases S/Node Coverage

E<sub>i</sub> = Individual test case (A cuckoo egg).

Before initiating the algorithm, it needs some input data for assistance. The user needs to input values for N, M, t and e. These values are used by the algorithm, and might vary from program to program.

With the initial input values, a test case that satisfies the constraints set by input values is randomly generated. Say, the test case consists of 3 eggs - E<sub>1</sub>, E<sub>2</sub> & E<sub>3</sub>.

*Step 1:* Generate a new random egg E<sub>i</sub> using levy flight.

*Step 2:* Select at random, an egg from I (say E<sub>2</sub>).

If  $n(\{E_1, E_2, E_3\}) < n(\{E_1, E_i, E_3\})$

Replace E<sub>2</sub> by E<sub>i</sub>. Else, no operation is carried out.

*Step 3:* Generate a new egg, say E<sub>j</sub> again using levy flight.

levy flight is used to prevent the case, where the new eggs generated also belong to the subset or branch of statements/nodes, which are already traversed or covered. Hence, levy flight allows us to move in random directions thus providing us with a possibility to cover all nodes.

Let  $n(\{E_j, E_i, E_3\}) = n_1$

Let  $n(\{E_1, E_j, E_3\}) = n_2$

Let  $n(\{E_1, E_i, E_j\}) = n_3$

*Step 4:* Plot number line between 0 & 1, for n<sub>1</sub>, n<sub>2</sub> and n<sub>3</sub> in terms of probabilities.

The number line is plotted rather than directly taking the value, which is greater from among the above n<sub>1</sub>, n<sub>2</sub>, n<sub>3</sub> so as to introduce randomness into this selection. This ensures that there is a high probability that the egg with greater value of nodes covered will be selected. However, because the number generated is random, there is no assurance of this. This is precisely the uniqueness of Cuckoo Search that plays a major role in reaching the global optimum without giving too much importance to the locally optimum solution.

*Step 5:* Generate a random number in the range (0, n<sub>1</sub>+n<sub>2</sub>+n<sub>3</sub>). According to the range in which the random number falls, select the corresponding test case for the next step.

*Step 6:* At the end of the above steps, I contain a set of the current test cases. Check if the test case is able to cover all nodes in the program i.e. if  $n(I) == M$ . If so then stop the algorithm.



If the node coverage of the new test case is less than that in the good tabu list i.e. if  $n(I) \leq n(T)$ , and if tabu life-span is expired i.e. if  $t=0$ , then revert back to the test case in good tabu list, and also reset tabu life-span.

If the node coverage is better than that in the good tabu list, i.e.  $n(I) > n(T)$ , but is less than the total number of nodes, then add a new test case to the good tabu list i.e.  $T=I$ .

If evolution life-span expires i.e. if  $e=0$ , then revert back to the test case in the good tabu list i.e.  $I=T$ . Also, increase the egg count by 1 and set evolution life-span back to the initial value.

Go to *Step 1*.

Continue the algorithm until the optimal test case is reached, which covers all the nodes. Since, we limited the total number of iterations to the evolution life-span, it will not infinitely keep trying with the same number of test cases, but will keep increasing the test cases every time the evolution life-span is reached. But it might also happen very infrequently that the algorithm might keep increasing the number of eggs for a very long time, but still doesn't find the optimal test case. In that case, it is up to the user to stop the algorithm when a manually set time limit expires.

The algorithm correctly works for numeric data (i.e., for generating test cases for numeric data types). However, for loop problems, the algorithm, if applied without any change, would take a lot more time than anticipated. It would execute the loop for the total number of times specified which is not required. The only aim is to generate test case(s) which would cover all nodes (i.e. statements) inside the loop, which might have been completed in the first iteration itself. But, the algorithm will continue to execute the loop till it ends. Hence, another variant of this algorithm is proposed to be used in case of loop problems.

For loops, a global flag is maintained which is unset by default. The loop will be executed (or the control will enter into the loop) only if the flag is unset. If at any point of execution of the algorithm all statements or nodes inside the loop are covered (or already traversed), then the flag will be set. Hence, when the loop is about to run for the next iteration, it will check for the flag (which has been set) and hence control will quit/break from the loop, thus preventing the redundant execution of loop statements for the rest of the iterations.

The proposed algorithm can be applied to problems concerning strings as well. However, the random generation of test cases using levy flight is not straightforward in this case. The next paragraph explains the representation of cuckoo eggs for string inputs as well as the modified algorithm to generate automated optimal test cases for the same.

Initially, the length(s) of the string input(s) and the ASCII range of character(s) that the string may contain are taken as inputs from the user. Every character in the string is treated as an individual egg according to the algorithm proposed earlier. Therefore, the number of eggs for each string input is equal to the total length of the string. At the beginning of the algorithm, each string input test case is randomly generated by generating the cuckoo eggs for each character in each of these strings using levy flights. This will be followed by the execution of the algorithm in the same way as proposed. Finally, every final egg that corresponds to a string input is converted back into its character value to obtain each optimal string test case.

### 4 Experimental Study

For experimentally studying the effects of the proposed algorithm, the classical triangle problem is chosen. This problem takes in three inputs that are to be the lengths for the sides of a triangle. It then prints the type of the triangle, which may be equilateral if all the sides are equal, isosceles if any two sides are equal and plain if none of the above hold true. The code for the same is shown in Figure 2. The Control Flow Graph (CFG) for the code and control flow graph is shown in Figure 2 and 3. Now, the working of the algorithm is illustrated through an example of the test case generation for the triangle problem. The values of the variables as discussed in the preceding section are set as follows:-

M = Total number of nodes in CFG = 13

N = Current number of input test cases = 3

t = Tabu life-span = 20 (This value is obtained by observing a number of experimental values and choosing the value for which the solution reaches faster)

e = Evolution life-span =1000 (This value is chosen to be high i.e. >500 if the expected number of test cases for a program is low, which in the triangle problem case is 3. It is chosen to be low i.e. in the range of 100 to 500 if the expected number of test cases is high).

```

int triType(int a, int b,
int c){
1   int type=PLAIN;
1   if (a<b)
2     swap(a,b);
3   if (a<c)
4     swap(a,c);
5   if (b<c)
6     swap(b,c);
7   if (a==b){
8     if (b==c)
9     type=EQUILATERAL;
10    else
11    type=ISOSCELES;
11  }
12  else if (b==c)
13    type=ISOSCELES;
13  return type;
}
    
```

Fig. 2. Code for the triangle problem

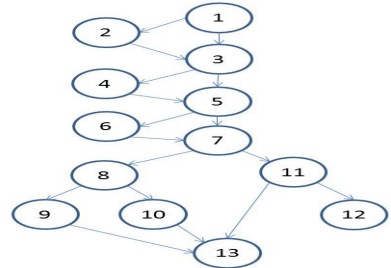


Fig. 3. CFG for the triangle problem

Here, each individual test case is of the form (a, b, c) denoting the values of a, b and c that are passed as inputs to the program. The range of test cases is set to 0-10.

Initial set of test cases is randomly generated using levy flight, say I<sub>1</sub>(3,9,3), I<sub>2</sub>(9,4,0) and I<sub>3</sub>(6,6,1).

Initial set of T = null set i.e. n(T) = 0.

n(I<sub>1</sub>,I<sub>2</sub>,I<sub>3</sub>) = 10 = Number of nodes covered.

*Start of Iteration*

Step 1: A new random egg is generated using levy flight, say C<sub>1</sub>(4, 2, 4).

Step 2: Now, one of I<sub>1</sub>, I<sub>2</sub> and I<sub>3</sub> is randomly chosen. Say, I<sub>1</sub> is selected.

n(C<sub>1</sub>,I<sub>2</sub>,I<sub>3</sub>) = 9 < 10 = n(I<sub>1</sub>,I<sub>2</sub>,I<sub>3</sub>) Hence, not replacing.

Step 3: A new random egg is generated using levy flight, say C<sub>2</sub>(4,7,8).

Now, n(C<sub>2</sub>,I<sub>2</sub>, I<sub>3</sub>) = 10; n(I<sub>1</sub>,C<sub>2</sub>, I<sub>3</sub>) = 12; n(I<sub>1</sub>,I<sub>2</sub>, C<sub>2</sub>) = 10

Step 4: Now, a number line is plotted from 0 to 10+12+10=32.

Step 5: A random number is generated, say 1.

Therefore, selected egg for replacement is  $I_1$ .

The new test case set is  $C_2, I_2, I_3$ .

Step 6: Now,  $n(I) = n(\{C_2, I_2, I_3\}) = 10$  is not equal to  $M=13$ . i.e. the no. of nodes covered by the new test case is not equal to the total number of nodes present in the program.

But,  $n(I) > n(T)$ , i.e. the no. of nodes covered by the new test case is more than the test case present in the good tabu list.

Therefore we add the new test case to the good tabu list, and  $T=I = (C_2, I_2, I_3)$ .

$n(T)=10$  i.e. reset the tabu life-span.

End of Iteration

When the termination condition reaches, the test cases present in I are the minimal number of test cases that are required in order to traverse all the nodes.

The results of applying the above algorithm to generate test cases for the triangle problem for different input ranges are tabulated in Table 1. The corresponding results for genetic algorithm were obtained from [9]. The graph for the same is shown in Figure 4. The proposed algorithm performs comparably for smaller test case ranges as observed in Table 1. However, for large test case ranges (i.e. more than 15), the proposed algorithm performs much better in comparison to the best existing search based optimization algorithm i.e. Genetic Algorithm [9].

## 5 Conclusion

This paper presents a heuristic method for automation of test data generation, using Cuckoo Search along with levy flights and Tabu Search, based on the number of nodes covered from among the total number of nodes that are present in the given program/software.

The results from the Experimental study as explained above clearly support this fact that the proposed algorithm performs better than the existing GA based strategy, which is depicted in figure 4. Though, the strategy for the string data is laid out, the application of the proposed approach for string input is yet to be explored through experimental comparison with other algorithm data.

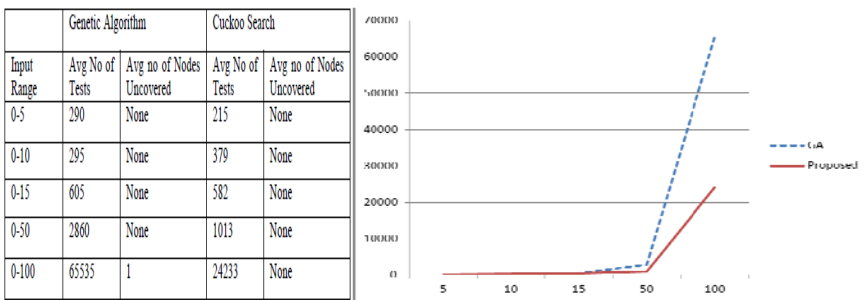


Fig. 4. Iteration (tabular representation) and Graph between Genetic Algorithm/GA (Dotted Line) and proposed algorithm (Solid Line) comparing the test case generation behavior

## References

1. Pressman, R.S.: Software Engineering: A Practioner's Approach, 6th edn., International edn., pp. 33–39. McGraw-Hill (2005)
2. Tahbaldar, H., Kalita, B.: Automated Test Data Generation: Direction of Research. *International Journal of Computer Science & Engineering Survey* 2(1) (February 2011)
3. Lakhotia, K., McMinn, P., Harman, M.: Automated Test Data Generation for Coverage: Haven't We Solved This Problem Yet?. In: *Testing: Academic and Industrial Conference - Practice and Research Techniques, TAIC Part 2009*, pp. 95–104 (2009)
4. Xanthakis, S., Ellis, C., Skourlas, C., Gall, A.L., Katsikas, S., Karapoulos, K.: Application of Genetic Algorithms to Software Testing. In: *Proceedings of the Fifth International Conference on Software Engineering and its Applications*, pp. 625–636 (1992)
5. Glover, F.: *Tabu Search fundamentals and uses*. University of Colorado, Notes for the Graduate School of Business (1994)
6. Srivastava, P.R.: Automated Software Testing Using Metaheuristic Technique Based on An Ant Colony Optimization. In: *Proceedings of International Symposium on Electronic System Design*, pp. 235–240. IEEE Explore (2010)
7. Beizer, B.: *Software Testing Techniques*, 2nd edn. Van Nostrand Reinhold (1990)
8. Yang, X.-S., Deb, S.: Cuckoo Search via Levy flights. In: *Proceedings of World Congress on Nature & Biologically Inspired Computing*, pp. 210–214. IEEE Publications, USA (2009)
9. Shen, X., Wang, Q., Wang, P., Zhou, B.: Automatic generation of test case based on GATS algorithm. In: *IEEE International Conference on Granular Computing, GRC 2009*, pp. 496–500 (2009)

# Selection of GO-Based Semantic Similarity Measures through AMDE for Predicting Protein-Protein Interactions

Anirban Mukhopadhyay<sup>1</sup>, Moumita De<sup>1</sup>, and Ujjwal Maulik<sup>2</sup>

<sup>1</sup> Department of Computer Science and Engineering, University of Kalyani, Kalyani-741235, India

anirban@klyuniv.ac.in, moumita.de2013@gmail.com

<sup>2</sup> Department of Computer Science and Engineering, Jadavpur University, Kolkata-700032, India  
umaulik@cse.jdvu.ac.in

**Abstract.** Protein-protein interactions (PPI) form the core part of the entire interatomic system for all the living elements. In this article, the role of different Gene Ontology(GO)-based semantic similarity measures in predicting PPIs have been explored. To find out a relevant subset of semantic similarity measures, a feature selection approach is developed with Angle Modulated Differential Evolution(AMDE), an improved binary differential evolution technique. In this feature selection approach, SVM classifier is used as a wrapper where different metrics like sensitivity, specificity accuracy and Area Under Curve (AUC) are measured to find the best performing feature subset. Results have been demonstrated for real-life PPI data of yeast.

**Keywords:** Protein-Protein Interaction, GO-based protein similarity, Angle Modulated Differential Evolution (AMDE), feature selection.

## 1 Introduction

Proteins are biological molecules responsible for what a living being is and does in physical sense [10]. Protein-protein interactions (PPIs) are necessary for carrying out most of the biological functions. Therefore it is very important to devise suitable methods for predicting new PPIs. In this article, the roles of traditional gene ontology (GO)-based taxonomic similarity measures like Resnik, Lin, Jiang-Conrath, Relevance, Cosine, Kappa etc. have been explored in predicting PPIs. Existing yeast PPI data set has been used as sample data.

Biologists are interested in measuring the functional similarity between protein pairs. As the characteristic of protein can be described with a set of GO terms so GO term similarity measure will provide a measure of functional similarity between two proteins. This article explores the possibilities of using GO-based functional similarity measures for predicting the interactions among proteins. Although there are a multitude of methods for computing GO-based similarities among proteins, not all of them are good measures for predicting the PPIs. In

order to deduce a well performing subset of similarity measure approaches, Angle Modulated Differential Evolution (*AMDE*) [11] is applied as a feature selection algorithm with SVM classifier as wrapper. To perform a complete analysis, each of the semantic similarity measures has been applied for the three separate Direct Acyclic Graph(DAG) of GO terms (biological process, molecular function and cellular component). Moreover, a novel intuitive approach for generating the non-interacting protein pairs has also been proposed.

## 2 GO-Based Semantic Similarity

In this section different taxonomy based semantic similarity measures that are used to measure the protein pair semantic similarity are briefly discussed.

Gene Ontology (GO) [1] is a tool for describing gene products depending on their functions. A set of GO terms are used to describe properly the functionality of a protein. GO terms are shared between proteins and this shared annotation is considered as the semantic similarity between proteins. Semantic similarity measure between protein pairs can be classified into two methods: edge-based and node-based. In case of edge based method for each protein an induced graph is constructed that consists of the GO terms associated with the protein and their parents. Semantic similarity in this approach is measured through the similarity of the induced graphs of two proteins in a given ontology. For node based methods semantic similarity is measured by comparing the similarities between the common ancestors or descendants of the input terms. Similarity between information content is the measure of semantic similarity of protein pairs. Information content is the occurrence of a term or its parent in the protein annotation data set. A rarely occurring term is considered to be more informative. The following are the different semantic similarity measures considered in this article.

### 2.1 Taxonomy-Based Semantic Similarity Measures

Resnik [9] proposed a method of semantic similarity measure where all the terms occurring in a taxonomy are considered as a *concept*( $C$ ). The information content of the concept  $C$  is formulated by the negative *log likelihood*  $-\log p(c)$ , where  $p(c)$  is the probability of encountering a concept. From the quantifier of information content it can be intuitively said that: as probability of a concept increases its information content decreases. According to Lin [5] every object is considered to belong to a class of taxonomy. Similarity between two objects is the similarity between those two classes. Edge-based semantic similarity measure is proposed by Jiang and Conrath [4] where information content is used as a decision factor to determine the strength of a link. Relevance approach [13] combines Lin and Resnik similarity measure. Some other taxonomy based similarity measures used in this paper are Cosine's Measure, Kappa's Measure [2], Czekanowski-Dice's Measure and Weighted Jaccard Measure [6].

## 2.2 Graph-Based Similarity Measure [3]

Couto et al. proposed a new of similarity measure based on Graph-based Similarity Measure (GraSM). The main objective of this approach is to find out the most informative common disjunctive ancestors. Common ancestors are said to be disjunctive if there is an independent path between the concept and the ancestor *i.e.* there must be at least one term of the ontology in the path that does not exist in the other paths. To calculate the similarity between two given terms GraSM similarity measures select all of their common disjunctive ancestors. Here the Graph-Based Similarity measure is applied for three different similarity measures as Resnik, Lin and Jiang-Conrath measure. Those are named as ResnikGraSM, LinGraSM, JiangConrathGraSM.

## 2.3 Avg, Max, Rcmx [7]

The similarity measures matrix  $S$  can be calculated using any of the similarity measure mentioned above viz. Resnik, Lin, Jiang-Conrath, Relevance, ResnikGraSM, LinGraSM, Jiang-ConrathGraSM. The package csbl.go provides three different measures of calculating protein pair similarity from the matrix  $S$  [7].

$$sim_{max}(A, B) = \{s_{ij}\}, \quad (1)$$

where,  $s_{ij}$ , is the maximum value of the matrix  $S$ ,

$$sim_{avg}(A, B) = \frac{\sum_{i=1}^n \sum_{j=1}^m s_{ij}}{n \times m}, \quad (2)$$

$$sim_{rcmax}(A, B) = \max\{rowScore, columnScore\}, \quad (3)$$

where

$$rowScore = \frac{1}{n} \sum_{i=1}^n \max_{1 \leq j \leq m} s_{ij}, \text{ and}$$

$$columnScore = \frac{1}{m} \sum_{j=1}^m \max_{1 \leq i \leq n} s_{ij}$$

## 3 Feature Selection Approach

This section describes AMDE and how it is applied for feature selection.

### 3.1 Classical AMDE [8]

Angle Modulated Differential Evolution (AMDE) is a discrete binary implementation of Differential Evolution. Let,  $f(x)$  be the function under consideration then at the initial stage a set of  $p$  possible solutions are randomly generated, called chromosome string in DE. If the function  $f(x)$  consists of  $n$  number of

parameters then each of the chromosome strings must be  $n$  dimensional. In case of binary DE each of the  $n$  chromosome elements are represented with an  $m$  bit binary string, so that the length of the chromosome string will be  $m \times n$ .

Angle Modulation is a technique of generating a bit string of the evolving population with a trigonometric function. The function is as follows (Equation: 4):

$$g(x) = \sin(2\pi(x - a) \times b \times \cos(A)) + d \quad (4)$$

where,

$$A = 2\pi \times c(x - a),$$

and  $x$  is a single element that belongs to the continuous valued chromosome string. If  $g(x)$  is greater than 0 then  $x$  is replaced with 1 and for the else condition, the value of  $x$  is 0. The initial population is then evolved to obtain the optimized solution  $X^*$ . A new offspring  $O_n$  is generated from three randomly chosen individual ( $C_{n1}$ ,  $C_{n2}$ ,  $C_{n3}$ ) of present generation. This is called mutation. Decision for mutation is taken through an uniformly chosen random number  $U(0,1)$  lies between 0 and 1. For each bit in the chromosome string  $j = 1, 2, \dots, n$  if  $U(0, 1) < p_r$  or,  $i = j$  then

$$O_n = C_{n3,j} + F(C_{n2,j} - C_{n1,j})$$

else,

$$O_{n,j} = C_{n,j}$$

where  $p_r$  is called crossover probability,  $i$  is a random number generated in domain of chromosome length and  $F$  is called mutation factor. If the value selected for  $p_r$  is large then many bits will go through the adverse change called mutation and vice versa. In case of AMDE the crossover and mutation function is performed on the 4-dimensional trigonometric function parameter tuple ( $a$ ,  $b$ ,  $c$ ,  $d$ ). More specifically AMDE reduces the  $n$  bit chromosome stream in 4 bit tuple, for evolution. This 4 bit tuple is used for generating  $n$  bit chromosome string (Equation: 4). Fitness for the offspring is calculated with the newly generated  $n$ -bit string. The child bit stream replaces its parent if fitness of child stream is better. With this iterative method AMDE population converges towards an optimal value but convergence to the global optimum is not guaranteed.

### 3.2 Feature Selection with AMDE

A set of 75 different semantic similarity measures are used here to predict PPIs. So AMDE population is implemented with individual chromosome string of length 75 bits, where each bit corresponds to a protein pair similarity measure. A feature is considered to be selected if its corresponding bit string position indicates 1. For each individual chromosome string SVM classification (5-fold cross validation over training set) is performed with the corresponding selected feature subset. Classification accuracy is noted as the fitness of that individual. At each generation the best individual and its corresponding feature subset is recorded. The set of features showing best fitness value at the final generation is taken as the selected feature subset.



## 4 Proposed Study

This section describes the steps followed in the proposed study.

### 4.1 Preparing the PPI Data Set

A set of 5000 *S. cerevisiae* proteins and 22209 interacting protein pairs have been retrieved from the Database of Interacting Proteins (DIP) (<http://dip.doe-mbi.ucla.edu/dip/Main.cgi>). Out of these proteins 4712 proteins and 21532 interacting protein pairs are used depending on their availability of gene annotation data. A set of 150000 number of possibly non-interacting protein pairs is randomly generated from the available protein list.

### 4.2 Preparing Non-interacting Protein Pairs

The performance of a classifier heavily depends on the quality of the negative samples (non-interacting protein pairs). Here a set of negative samples is generated based on the assumption that the protein pairs, those do not have common CC terms can be assumed to have a minimum chance of interaction. Since these pair of proteins do not reside at the same part of the cell, there is less possibility for coming in contact with each other for interaction. The number of such protein pairs is nearly 99000. These 99000 protein pairs are sorted in descending order of the summation of the number of cellular component related GO-terms present in those protein pairs. From this sorted list 21530 protein pairs are taken for further experiment.

### 4.3 Protein Pair Semantic Similarity Measure

To measure the semantic similarity between protein pairs we have used csbl.go (<http://csbi.ltdk.helsinki.fi/csbl.go/>) [7], a R (<http://cran.r-project.org/>) package for computing semantic similarity using Gene Ontology. Protein pair semantic similarity is measured using 8 different semantic similarity measures viz. Resnik, Lin, JiangConrath, Relevance, Cosine, Kappa, CzekanowskiDice, WeightedJaccard. Graph based semantic similarity is considered for only 3 of the above methods viz. Resnik, Lin, JiangConrath named as ResnikGraSM, LinGraSM and JiangConrathGraSM respectively. The methods like Resnik, Lin, JiangConrath, Relevance and ResnikGraSM, LinGraSM and JiangConrathGraSM is used to evaluate semantic similarity between the three different gene ontology (Biological Process (BP), Cellular Component (CC), and Molecular Function (MF)) categories separately. So for these 7 methods we will get ( $7 \times 3 = 21$ ) different measures. Now for all of the methods discussed above there are three different measures: average(avg), maximum(max) and row-column maximum(rcmax). Therefore we are considering a total of  $(21 \times 3 + 4 \times 3) = 75$  features. Now the final data set is created that contains protein pairs and their 75 different semantic similarity measures in each row.

#### 4.4 Feature Selection

In order to reduce the protein interaction dataset used for feature selection, a k-means clustering algorithm is applied on 21530 interacting and non-interacting protein pairs to get 100 cluster centers for each sample set. These 200 cluster centers (100 interacting and 100 noninteracting) with 75 features are used as the training set for feature selection through AMDE.

AMDE is implemented with the population of size 40 and chromosome length 75. Initially the chromosome strings are populated with the bit string generator function (Equation: 4). For each chromosome string the temporarily created protein interaction dataset with 200 data points and selected features is classified with support vector machine (SVM) [12] classification (5-fold cross validation on the training set). Fitness of an individual chromosome is the classification accuracy. The subset of features that gives optimal value of fitness at the final generation is used for classifying the original protein interaction dataset of 21530 interacting and 21530 non-interacting protein pairs.

### 5 Results and Discussion

In this section the classifier output with their corresponding input varieties are discussed. At the final generation the reduced protein interaction dataset has given an optimized fitness value of 96 with 9 selected similarity measures. These 9 selected features are: Resnik-CC-max, Resnik-CC-rcmax, Resnik-MF-rcmax, Resnikgrasm-BP-avg, Resnikgrasm-CC-avg, Lin-CC-max, JiangConrath-CC-avg, JiangConrathgrasm-BP-avg and CzekanowskiDice-CC.

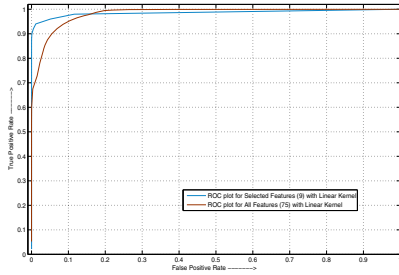
The complete protein interaction dataset with 21530 interacting and 21530 possible non-interacting is then classified with SVM classifier using that selected subset of 9 semantic similarity measures. A comparative study based on like sensitivity, specificity and accuracy for both the dataset consisted with all the 75 features and for selected subset of 9 features are reported in Table 1. It is evident from the table that the sensitivity, specificity and accuracy are consistently better for the selected feature set as compared to that for all features. Moreover, it can be noticed that RBF kernel performs better than the linear and the polynomial kernels of SVM.

Figure 1 shows the Receiver Operating Characteristic (ROC) plots between true positive rate (TPR) and false positive rate (FPR) for both the complete

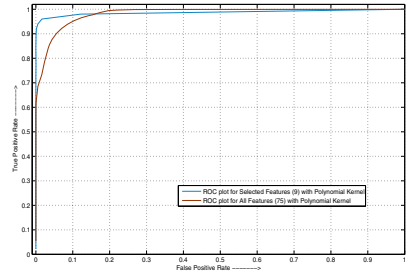
**Table 1.** Classification results for different feature set

Kernels	All Features (75)			Selected Features (9)		
	Sensitivity	Specificity	Accuracy	Sensitivity	Specificity	Accuracy
Linear	87.0135	99.8096	93.4115	94.0641	98.8017	96.4329
Polynomial	64.9837	99.8978	82.4408	94.8769	99.0571	96.967
RBF	98.5555	88.485	93.5207	98.7552	96.8602	97.8077

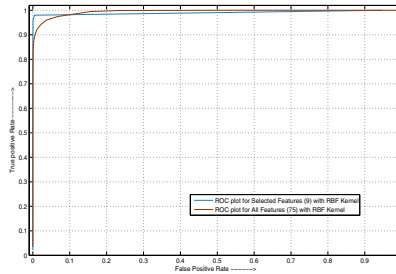
feature set and selected subset of 9 features for all the three different kernels. It appears from the figure that for all the three kernels, the plots corresponding to the selected features are closer to the upper left corner (TPR=1, FPR=0) of the figures indicating the better performance of the selected features. This is also established from the area under curve (AUC) values reported in Table 2. It is clear from the table that the subset of 9 selected features is consistently giving higher AUC values than the complete feature set except for the AUC value obtained using Polynomial kernel.



(a)



(b)



(c)

**Fig. 1.** Comparative study on ROC plots using different Kernels (a) Linear Kernel, (b) Polynomial Kernel and (c) Gaussian (RBF) Kernel

**Table 2.** AUC values

Kernels	All Features (75)	Selected Features (9)
<b>Linear</b>	0.9822	0.9862
<b>Polynomial</b>	0.9871	0.979
<b>RBF</b>	0.9822	0.9944

## 6 Conclusion

In this article, we have explored the utility of GO-based semantic similarity measures in predicting protein-protein interactions in yeast. Out of 75 different semantic similarity measures, a set of 9 measures have been selected through angle-modulated DE-based feature selection which uses SVM classifier as wrapper. The performance of the selected feature set has been demonstrated based on selectivity, sensitivity, accuracy and ROC analysis. The RBF kernel of SVM has in general been found to perform better than the linear and polynomial kernels. As a future work, the performance of other feature selection methods can be studied and compared with AMDE-based method. Moreover the biological relevance of the predicted PPIs is to be measured.

## References

1. Ashburner, M., Ball, C., Blake, J., Botstein, D., Butler, H., Cherry, J., Davis, A., Dolinski, K., Dwight, S., Eppig, J., et al.: Gene ontology: tool for the unification of biology. the gene ontology consortium. *Nature Genetics* 25, 25–29 (2000)
2. Benabderrahmane, S., Smail-Tabbone, M., Poch, O., Napoli, A., Devignes, M.-D.: Intelligo: a new vector-based semantic similarity measure including annotation origin. *BMC Bioinformatics* 11(588), 1471–2105 (2010)
3. Couto, F.M., Silva, M.J., Coutinho, P.M.: Measuring semantic similarity between gene ontology terms. *Data and Knowledge Engineering* 61(10), 137–152 (2007)
4. Jiang, J.J., Conrath, D.W.: Semantic similarity based on corpus statistics and lexical taxonomy. In: *Proc. International Conference Research on Computational Linguistics*, Taiwan (1997)
5. Lin, D.: An information-theoretic definition of similarity. In: *Proc. 15th International Conference on Machine Learning*, San Francisco, CA, pp. 296–304 (1998)
6. Mukhopadhyay, A., Maulik, U., Bandyopadhyay, S., Brors, B.: Goga: Go-driven genetic algorithm-based fuzzy clustering of gene expression data. In: *Proc. Int. Conf. Systems in Medicine and Biology (ICSMB 2010)*, pp. 349–353 (2010)
7. Ovaska, K., Laakso, M., Hautaniemi, S.: Fast gene ontology based clustering for microarray experiments. *BioData Mining* 1(11) (2008)
8. Pampar, G., Engelbrecht, A., Franken, N.: Binary differential evolution. In: *Proc. IEEE Congress on Evolutionary Computation*, pp. 16–21 (2006)
9. Resnik, P.: Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research* 11, 95–130 (1999)
10. Setubal, J., Meidanis, J.: *Introduction to Computational Molecular Biology*. PWS Publishing Company, MA (1997)
11. Storn, R., Price, K.: Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization* 11(4), 341–359 (1997)
12. Vapnik, V.: *The Nature of Statistical Learning Theory*. Springer, Heidelberg (1995)
13. Wang, J.Z., Du, Z., Payattakool, R., Yu, P.S., Chen, C.-F.: A new method to measure the semantic similarity of go terms. *BMC Bioinformatics* 23(10), 1274–1281 (2007)

# Towards Cost-Effective Bio-inspired Optimization: A Prospective Study on the GPU Architecture

Paula Prata<sup>1,2</sup>, Paulo Fazendeiro<sup>1,2</sup>, and Pedro Sequeira<sup>1</sup>

<sup>1</sup> University of Beira Interior, Department of Informatics, Portugal

<sup>2</sup> Institute of Telecommunications (IT), Portugal

pprata@di.ubi.pt, fazendeiro@ubi.pt, morps@live.com.pt

**Abstract.** This paper studies the impact of varying the population's size and the problem's dimensionality in a parallel implementation, for an NVIDIA GPU, of a canonical GA. The results show that there is an effective gain in the data parallel model provided by modern GPU's and enhanced by high level languages such as OpenCL. In the reported experiments it was possible to obtain a speedup higher than 140 thousand times for a population's size of 262 144 individuals.

**Keywords:** GPGPU, Parallel Genetic Algorithms, OpenCL, data parallelism.

## 1 Introduction

Nowadays any desktop computer or notebook can be equipped with a multi-core Central Processing Unit (CPU) and a Graphics Processing Unit (GPU). If that heterogeneous processing power is fully exploited by programmers, application's performance can be improved by several orders of magnitude [1][2]. Parallelizing applications for these multi-core processors (CPU and GPU) is challenging because of the discrepancy between CPU and GPU existing processing models.

The basic idea to tackle an Evolutionary Algorithm (EA) in parallel consists in dividing the EA into chunks and solving the chunks simultaneously resulting in a sort of task parallelism, formalized in four main models: master-slave, fine-grained, coarse-grained and hierarchical [3]. From the set of open problems pointed by the author we highlight the determination of "an adequate communications topology that permits the mixing of good solutions, but that does not result in excessive communication costs". This is a problem that can probably be solved with the multi-core architectures that started to be explored for EAs in recent works. With the advent of General Purpose GPU (GPGPU) it is possible to use a new model of parallelism where the same operation can be performed across thousands of data elements simultaneously. In this work we try to take advantage of this new model.

The potential for acceleration of population based, stochastic function optimizers using GPUs has been already verified in a number of recent works (e.g. [4][5][6] just to mention a few) over a representative set of benchmark problems. How the GPUs speedups are related with the dimension of the problem remains an open question, particularly when there is a need to use the main memory.

In this paper the canonical Genetic Algorithm was studied, the set of operations that could be parallelized was identified and a parallel OpenCL implementation for an NVIDIA GPU was developed aiming at the exploration of the full potential of the GPU. With the aid of two well-know benchmark problems the impact of varying the population size and the dimensionality of the problem (genotype length) in the speedup of the parallel version was analyzed. In the reported experiments it was possible to obtain a maximum speedup higher than 140 000 times for a population size of 262 144 individuals.

In the remaining of this paper for the sake of self-containment, we begin by presenting a brief introduction to the genetic algorithms. Next we introduce the programming model of the OpenCL framework and describe the architecture of the used GPU. In section 4 we describe the details of the implementation and present the obtained results. Finally in the conclusions section we discuss the presented results and pinpoint some directions for future work.

## 2 Genetic Algorithms

The Genetic Algorithms (GAs) are adaptive robust methods with application in search, optimization and learning problems [7-12]. As their name suggests, were inspired by the principles of genetics and natural evolution of biological organisms. They are based on the observation of the genetic processes essential for a natural population to evolve and their consequent translation to the environment of the specific problem at hand. Its strength stems from the inherent capability to find the global optimum in a multi modal search space. The basic principles of GAs were established rigorously by Holland in 1975 [13]. The canonical GA is briefly specified in Algorithm 1.

---

### **Algorithm 1.** Canonical Genetic Algorithm

---

*Initialize the population*

*Evaluate the fitness of the individuals*

*Repeat the following steps until the termination criterion has been met.*

*Step 1 – Select the solutions for the matting pool.*

*Step 2 – Generate new offsprings performing crossover and mutation.*

*Step 3 – Evaluate the fitness of the new individuals.*

---

For each problem it is necessary to establish beforehand an adequate representation or codification of solutions and a fitness function to grade them. During the execution of the algorithm the parents are selected for reproduction and from their recombination are produced new offsprings. The evolution of the potential solutions over successive generations comprises different phases. Generally speaking, the first phase involves the quantitative evaluation of each individual in the population. This value determines the probability that an individual has to be selected and to carry its genetic material for the next phase. In the second phase, the selected individuals (potential solutions) are given the chance to mate and exchange genetic material with

other individuals by means of a crossover operator. The result of this reproduction phase is a new offspring population, which replaces (or sometimes compete with) the previous population. Some of these newly born individuals were possibly prone to some mutations. This process continues until a stop criterion has been met.

The GAs do not assure the optimal global solution for a problem but usually can find sufficiently good solutions in a sufficiently fast way. In fact, GAs demonstrate its usefulness in complex combinatorial problems or the ones with solutions defined on multi-modal surfaces and lacking specialized techniques (more accurate and faster than GAs). GAs require a limited amount of knowledge about the problem being solved: relative evaluation of the candidate solutions is enough and no derivatives of cost functions are required. This can be a definitive advantage when compared with other candidate methods of optimization; whenever the derivatives of the involved functionals are computationally demanding or the search space has no continuity [14].

### 3 The OpenCL Language and GPU Architecture

OpenCL is an open standard for general-purpose parallel programming, across multi-core CPUs, GPUs and other processors [18]. In relation to other interfaces for the GPU, as CUDA (Compute Unified Device Architecture) from NVIDIA [16], or Brook+ from AMD/ATI [17], it has the advantage of being platform-independent. In the parallel programming model of OpenCL, an application runs on a platform that consists of a host connected to one or more OpenCL devices [19]. An OpenCL device is divided into one or more compute units, which are further divided into one or more processing elements, or cores. An application consists of a host program that executes on the host, capable of launching functions (kernels) that execute in parallel on OpenCL devices.

When a kernel is submitted for execution by the host, it is defined an index space. The same kernel can be executed simultaneously by a high number of threads, each one for a point in the index space. Each thread, that is, each kernel instance, is called a work-item and is identified by its point in the index space. The same code can be executed over different data items following a SIMD (Single Instruction Multiple Data) model. In that case we are implementing a data parallel programming model.

Additionally, work-items can be organized into work-groups. Several work-groups can be executed in a SPMD (Single Program Multiple Data) model. In that case, although all processing elements run the same kernel, each with its own data, they maintain their own instruction counter and the sequence of instructions can be quite different across the set of processing elements. A task parallel programming model in OpenCL can be defined when a single instance of a kernel is executed independent of any index space [19].

The NVIDIA GPU GeForce GTX 295 used in this work, based on the GT200 architecture, is built as an array of multithreaded streaming multiprocessors. Each multiprocessor consists of eight scalar processor cores, each one with a set of registers associated, and a common shared memory area of 16KB [16]. Double precision floating-point operations are performed by a double precision unit shared by the eight cores of each multiprocessor (just for devices with “compute capability” equal or greater than 1.3). Using the OpenCL terminology, a streaming multiprocessor is a compute unit, and a processor core is a processing element. When a kernel is

launched, the work-groups, and corresponding work items are numbered and automatically distributed by the compute units with capacity to execute them. Work groups are assigned to compute units, and the work items of each work group are executed on the processing elements of the compute unit. Work items of the same work group can communicate through the multiprocessor shared memory, which is called local memory in OpenCL.

Because the GPU just processes data stored in its memory, the program data must be copied to the global memory of GPU before executing the kernel. In the end the results must be copied back to CPU memory. Global memory can be accessed by any thread in the program, but has an access latency of 400-600 clock cycles. There exist two additional read-only global memory spaces, texture and constant. There is a total of 64KB constant memory on a device, which after the first access is cached (thus all subsequent requests for this element do not need to be read from global memory) resulting in an access time near to zero [20].

## 4 Experiments and Results

Here, we aim to address the following issues: i) Is there an effective gain resulting from adopting a GPU architecture to tackle an EA problem? Specifically, what is the level of problem complexity that makes a problem better suited to be solved in the GPU. ii) To what extent is the performance of the parallel algorithm dependent on the precision of the encoding used? Notwithstanding the actual GPUs double precision arithmetic improvement still there are observable differences in performance dependent on the choice of single or double precision. iii) How could the different memory spaces be explored in order to overcome the huge latency of global memory? This is particularly relevant, as the access to the different types of memory present in the GPU can heavily constrain the obtained speedups.

The parameters of the GA were kept constant in the presented experiments. Since the GAs are designed to maximize the merit function we transformed the minimization problems into problems of maximization through the transformation  $f = I/(I + J)$ , where  $J$  is the objective function to minimize. A stopping criterion of 1000 iterations was used. The selection operator applied in the experiments was stochastic sampling with replacement. In order to maintain a constant workload all the population individuals were subjected to the blend crossover (BLX-alpha) operator chosen due to its suitability to the real-valued chromosome encoding that was used [15]. The probability of mutation was equal to 0.025. Furthermore, to prevent good solutions from disappearing during the evolutionary process it was used an elitist approach maintaining the best solution. The choice of all the above numeric values of the parameters was based on an extensive set of preliminary experiments, with the speed of convergence rate being the main guiding mechanism.

The solution quality and obtained speedups of the implemented GA were analyzed using Rosenbrock's and Griewangk's functions [12] two artificial benchmark functions commonly used for GA analysis. Two similar versions of the canonical genetic algorithm were developed: a sequential one executed on the host CPU and a parallel one executed on the GPU. After a preliminary analysis of variance it was found no statistical difference between the solutions' quality of the two versions.



The used GeForce GTX 295 has 30 multiprocessors, each one with 8 cores (at 1.24 GHz), with a maximum of 512 threads per work group. The host machine is an Intel Core 2 Quad Q9550 at 2.83 GHz. The results for small population's sizes (100, 500 and 2500) were obtained as the average of 30 runs with 1000 iterations each. Due to time limitations, the corresponding results for bigger population's sizes ( $2^{16}$ ,  $2^{17}$  and  $2^{18}$ ) were obtained with only 10 iterations each (the very same setup would imply a total running time of more than 8 months on the CPU!). The maximum value of  $2^{18}$  for the population size was previously adopted in works that apply island models (e.g. [5] uses 1024 islands and 256 individuals per island).

Table 1, shows the CPU execution times of a single iteration, for the Rosenbrok and Griewangk's functions, varying the dimensionality of the problem (for 2, 5 and 15 genes) when the population is represented with single precision values, that is, with floats. As can be seen the times go from few milliseconds to about 5 minutes. For the biggest population, more than 3 days are needed to perform 1000 iterations in CPU.

**Table 1.** Single iteration execution times (milliseconds) for Rosenbrok and Griewangk's functions using single precision in CPU. Both the dimensionality of the problem (2, 5, 15) and the cardinality of the population vary.

Pop. size	Dimensionality (Rosenbrok)			Dimensionality (Griewangk)		
	2	5	15	2	5	15
<b>100</b>	0,13	0,25	0,66	0,17	0,32	0,82
<b>500</b>	1,47	2,11	4,11	1,67	2,40	4,88
<b>2500</b>	30,00	33,13	43,15	30,61	34,22	47,28
<b>65536</b>	20 212,14	20 219,93	20 438,43	20 228,13	20 266,90	20 437,19
<b>131072</b>	80 464,52	80 929,15	81 485,45	81 123,55	80 998,96	81 124,00
<b>262144</b>	321 128,99	322 231,34	321 295,62	323 850,75	325 134,90	325 934,74

To build the parallel version for GPU, five functions of the sequential version are converted into kernels: 1- evaluation, 2 – roulette (partial sums), 3 – roulette (normalization), 4 - crossover and 5- mutation. As each kernel operates on the results of the previous one, kernel 2 is only launched after kernel 1 finish, and so on. The first population is generated in the CPU; the GPU evaluates it and generates the next population. For all but the initial population the data is copied to CPU after evaluation, to check for the convergence criteria and calculate the element with the best fitness.

The number of work items for each case studied corresponds to the size of the population,  $N$ , i.e. each kernel will be executed  $N$  times, each one over one population element. As our GPU has 30 multiprocessors we choose for the number of work items per work group the value of  $\lceil \sqrt{N/30} \rceil$ . Thus for a population of e.g. size 100, each work group has just 3 or 4 work items but there are no idled multiprocessors. As threads are scheduled in blocks of 32 it is considered a good practice to assign to each work group 32 work items, or a multiple of that value [20]. However this means that in the case of small populations, shall one choose to have work groups with 32 or more work items there would be several multiprocessors without any work. Instead we decided to distribute the workload between the available multiprocessors.

For populations of size bigger than 960 (30x32) each work group has more than 32 threads. Thus the capacity of the GPU is better used. From that dimension on, it is expected that more performance gains are obtained with the increasing ratio between the number of operations and the number of global memory accesses.

Table 2 presents the speedups ( $exec\_time\_CPU / exec\_time\_GPU$ ) obtained when running the parallel version on GPU, using single precision values and storing the population’s vector in global memory.

**Table 2.** Speedups on Rosenbrok and Griewangk’s functions depending on the size of the population and dimensionality of the problem, using global memory

Pop. size	Dimensionality (Rosenbrok)			Dimensionality (Griewangk)		
	2	5	15	2	5	15
<b>100</b>	0,63	1,05	1,82	0,85	1,41	2,74
<b>500</b>	1,99	2,70	4,41	2,52	3,25	5,76
<b>2500</b>	8,52	9,31	11,41	8,70	9,66	12,78
<b>65536</b>	48,61	48,64	48,19	48,65	48,80	48,39
<b>131072</b>	53,32	53,01	52,49	53,75	53,08	52,32
<b>262144</b>	55,03	54,86	53,99	55,34	55,35	54,93

As can be observed for the biggest populations the speedup is around fifty. Now, a single iteration on the biggest population needs less than 6 seconds. Accessing the global memory is the main bottleneck when working in GPU, because of the high access latency. When the complexity of the functions increases (from 2 genes to 15) we get a similar speedup. Thus the time spent in the access to global memory can accommodate much more calculations without increasing the execution time.

A second version of the algorithm was tested using constant memory (with much smaller latency) where possible, namely in the kernel 2 (roulette) and in kernel 4 (crossover). Within kernel 2 the vector that contains the population is copied to the constant memory and remains there until the end of the kernel 4. Table 3 presents the speedups obtained in GPU with constant memory, and using single precision values.

Now, an impressive improvement can be observed. Considering the biggest population size, with 2 genes, the speedup goes from 55 to more than 140 000. With

**Table 3.** Speedup on Rosenbrok and Griewangk’s functions depending on the size of the population and dimensionality of the problem, using constant memory

Pop. size	Dimensionality (Rosenbrok)			Dimensionality (Griewangk)		
	2	5	15	2	5	15
<b>100</b>	0,89	1,59	1,96	1,25	2,14	2,90
<b>500</b>	3,55	7,57	5,39	4,06	5,06	11,05
<b>2500</b>	17,27	37,21	121,77	17,50	14,68	192,58
<b>65536</b>	39 792,44	13 283,90	3 426,60	30 665,49	15 213,60	3 694,45
<b>131072</b>	79 694,61	27 080,05	7 097,09	66 343,32	24 067,07	7 638,30
<b>262144</b>	140 461,04	52 507,96	13 975,53	120 155,49	53 898,33	14 412,27

15 genes the speedup increases from 55 to around 14 000, which is also a very good result. Using constant memory it becomes visible that increasing the number of genes decreases the speedup.

These results show that with global memory the main limiting factor of performance is the time to access the memory.

A similar study was performed using double precision values. It was observed that for the case of global memory, in the worst scenarios the speedup values are very close to the ones for single precision values (around 60). Besides that, as with single precision, increasing complexity has no significant impact in the execution time. Memory access latency explains both situations. A very important factor of performance for the architecture of the GPU used is the coalescing of the accesses to the global memory. The best performance is obtained when all threads in the set that is scheduled for execution, in a processing element, access to data stored in contiguous memory positions [20]. Each memory transaction accesses to a segment of 64B thus the best situation occurs when the data needed by the set of threads is serviced with just one access. As the execution model in each processing element is SIMD, if different threads access data from different segments the execution is serialized and performance decreases.

When using the constant memory the speedups, as in the float case, are much better than the results with global memory and vary with the complexity of the problem. For the greatest dimension with 2 genes the speedup is almost 30 000 and with 15 genes the speedup is similar to the speedup in single precision, about 14 000. This points to that, with this dimension and this complexity, probably the limiting factor of performance remains the memory access latency.

## 5 Conclusions

The results obtained in the presented work show that there is a very effective gain in the data parallel model provided by modern GPU's and enhanced by high level languages such as OpenCL. The speedups obtained show that, when working with populations with hundreds of thousands elements, the parallel version can have similar solution quality while being thousands times faster than the sequential version. Varying the dimensionality of the problem has revealed that the main limiting factor of performance is the time to access the global memory. Identifying the steps of the algorithm where it is possible to use low latency memories, such as constant memory, was the key to improve the speedup by several orders of magnitude.

Working with doubles and global memory has shown that although in the GPU used just exist 30 double precision units (one per multiprocessor) the speedups are better than the ones obtained with floats. This builds evidence that further supports the hypothesis that memory is the main limiting factor of performance.

Left as future work is the improvement of the performance for higher dimensionalities, which can be studied by further parallelizing the kernel 4 (crossover). Using multi-dimensional index spaces, instead of having a single work item per individual, it is possible to have one work item for each gene of each individual. Memory accesses, besides using low latency memories, should also be explored in two other aspects that are left as future work. Firstly by avoiding costly data transfers between the device and the host - in the implemented algorithm the search for the best solution is done in CPU, parallelizing this step by a reduction

algorithm could improve performance. Secondly by improving the memory access pattern - designing the algorithm in such a way that it presents a pattern of memory access in which threads of the same scheduled unit access to data stored in contiguous memory positions could also improve significantly its performance.

## References

1. Woodward, P., Jayaraj, J., Lin, P.-H., Yew, P.-C.: Moving Scientific Codes to Multicore Microprocessor CPUs. *Comput. in Sci. & Engineering* 10(6), 16–25 (2008)
2. Feinbube, F., Troger, P., Polze, A.: Joint Forces: From Multithreaded Programming to GPU Computing. *IEEE Software* 28(1), 51–57 (2011)
3. Cantú-Paz, E.: A Survey of Parallel Genetic Algorithms. *Calc. Parallels* 10 (1998)
4. Zhou, Y., Tan, Y.: GPU-based parallel particle swarm optimization. In: *IEEE Congress on Evolutionary Computation, CEC 2009*, pp. 1493–1500 (2009)
5. Pospichal, P., Jaros, J., Schwarz, J.: Parallel Genetic Algorithm on the CUDA Architecture. In: Di Chio, C., Cagnoni, S., Cotta, C., Ebner, M., Ekárt, A., Esparcia-Alcazar, A.I., Goh, C.-K., Merelo, J.J., Neri, F., Preuß, M., Togelius, J., Yannakakis, G.N. (eds.) *EvoApplications 2010. LNCS*, vol. 6024, pp. 442–451. Springer, Heidelberg (2010)
6. de Veronese, L., Krohling, R.: Differential evolution algorithm on the GPU with C-CUDA. In: *IEEE Congress on Evolutionary Computation, CEC 2010*, pp. 1–7 (2010)
7. Coello, C., Van Veldhuizen, D., Lamont, G.: *Evolutionary Algorithms for Solving Multi-Objective Problems. Genetic Algorithms and Evolutionary Computation Series*, vol. 5. Springer, Heidelberg (2002)
8. Deb, K.: *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, New York (2001)
9. Goldberg, D.: *Genetic Algorithms in search, optimization and machine learning*. Addison-Wesley (1989)
10. Beasley, D., Bull, D., Martin, R.: An overview of genetic algorithms: Part 2, research topics. *University Computing* 15(4), 170–181 (1993)
11. Janikow, C., Michalewicz, Z.: An experimental comparison of binary and floating point representations in genetic algorithms. In: *Proc. of the Fourth International Conference in Genetic Algorithms*, pp. 31–36 (1991)
12. Bäck, T., Fogel, D., Michalewicz, Z.: *Handbook of Evolutionary Computation*. Institute of Physics Publishing Ltd., Oxford Univ. Press, Bristol, New York (1997)
13. Holland, J.: *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor (1975)
14. Valente de Oliveira, J.: Semantic constraints for membership function optimization. *IEEE Trans. on Systems, Man, and Cybernetics, Part A: Systems and Man* 29(1), 128–138 (1999)
15. Eshelman, L., Schaffer, J.: *Real-coded genetic algorithms and interval-schemata*, vol. 3, pp. 187–202. Morgan Kaufmann, San Mateo (1993)
16. NVIDIA Corporation: *NVIDIA CUDA Programming guide*, version 2.3.2 (2009)
17. Buck, I., Foley, T., Horn, D., Sugerman, J., Fatahalian, K., Houston, M., Hanrahan, P.: Brook for GPUs: stream computing on graphics hardware. In: *ACM SIGGRAPH*, pp. 777–786 (2004)
18. Khronos group: *OpenCL – The Open Standard Parallel Computing for Heterogeneous Devices* (2010), <http://www.khronos.org/opencl/>
19. Munshi, A. (ed.): *The OpenCL Specification Version: 1.1*, Khronos OpenCL Working Group (2011)
20. NVIDIA OpenCL Best Practices Guide, Version 1.0 (2009)

# Cricket Team Selection Using Evolutionary Multi-objective Optimization

Faez Ahmed, Abhilash Jindal, and Kalyanmoy Deb

Kanpur Genetic Algorithms Laboratory (KanGAL),  
Indian Institute of Technology, Kanpur,  
Kanpur, India  
{faez, ajindal, deb}@iitk.ac.in  
<http://www.iitk.ac.in/kangal>

**Abstract.** Selection of players for a high performance cricket team within a finite budget is a complex task which can be viewed as a constrained multi-objective optimization problem. In cricket team formation, batting strength and bowling strength of a team are the major factors affecting its performance and an optimum trade-off needs to be reached in formation of a good team. We propose a multi-objective approach using NSGA-II algorithm to optimize overall batting and bowling strength of a team and find team members in it. Using the information from trade-off front, a decision making approach is also proposed for final selection of team. Case study using a set of players auctioned in Indian Premier League, 4th edition has been taken and player's current T-20 statistical data is used as performance parameter. This technique can be used by franchise owners and league managers to form a good team within budget constraints given by the organizers. The methodology is generic and can be easily extended to other sports like soccer, baseball etc.

## 1 Introduction

Formation of a good team for any sports is vital to its success. Team selection in most sports is a subjective issue using commonly accepted notions to form a good team. In this work we have taken game of cricket as an example to demonstrate applicability of multi-objective optimization methodology to subjective issue of team formation from a set of players using available statistics. Cricket is a game played between two teams of 11 players where one team bats, trying to score as many runs as possible while the other team bowls and fields, trying to limit the runs scored by the batting team [1, 4]. Batting and bowling strength of a team are the major criteria affecting its success along with many other factors like fielding performance, captaincy, home advantage etcetera. We have explored the problem of building a 'good' team out of a set of players given the past performance statistics and suggested a new methodology from the perspective of multi-objective genetic optimization. Optimization studies have been done in many sports [6, 9, 11], and also has been done in various fields in cricket [10,12]. Just as in most league competitions a pool of players is provided as an input along with their performance statistics. Each player is paid a certain amount of

money by the team owners for playing for their team, which we refer to as player's cost. League organizers impose an upper limit on budget for each franchise/club to avoid giving undue advantage to rich franchises. Player cost is either fixed by organizers as salary, decided through auction or determined by some form of contract agreement. We have considered Indian Premier League (IPL) as a test case for our analysis. IPL is a professional league for T-20 cricket competition in India. As of now, not much literature is available for any team selection methodology in cricket. In IPL, the franchise managers have the task of building a good team within budget cap. Individual players are bought by the franchises during a public auction of the players. Since the total number of players in the market pool is large, the challenge of finding the optimal teams becomes increasingly complicated and common sense methods, mostly employed, may fail to give a good team. Data used for this work (uploaded on [2]) has a pool of 129 players from IPL 4th edition. We have used performance statistics of each player in international T-20. The need of an effective optimization technique can be justified by rough calculation of the size of the decision space. From the given data of only 129 players from IPL-4 auction, containing 10 captains and 15 wicket-keepers, the total number of possible teams under the constraints of at least one wicketkeeper and one captain is as follows

$$\text{Total Teams} = \binom{10}{1}C \binom{15}{1}C \binom{127}{9}C$$

Considering the large number of different possible team combinations (order of  $10^{15}$ ), finding optimal teams under the added constraint of budget is not trivial. Currently most of the team selections are done using different heuristics or greedy algorithms. Usually, two or three high performance batsmen or bowlers are picked and the remaining team slots are filled according to budget constraints. But, this approach may not always give an optimal solution since matches are won by team effort. In such scenarios, overall quality of team may be poor. For example, a team with the best bowlers in the world may not win due to their inability to chase even a small target due to their poor batting performance. Hence, our aim is to investigate formation of an overall optimal team.

## 2 Strategy and Optimization Methodology

In cricket, player statistics has multiple parameters like number of matches played, total runs made, strike rate, number of wickets taken, number of overs bowled etc. It is important to identify those statistical parameters which reliably indicate player's performance. The overall aim of a franchise is to build a team of 11 players with optimum bowling, batting as well as fielding performance within budget and rule constraints. Rule based constraints like presence of at least one player capable of wicket-keeping or maximum 4 overseas players in playing 11 also have to be taken in account. Considering the large amount of statistical data denoting various cricketing attributes that is available for each of the players, we first tend to reduce the dimension of data. One approach can be to use standard batting and bowling rating of cricketers obtained after exhaustive statistical analysis. Such ratings like the ICC

world cricket rating [5], takes into account multiple factors of performance. But, such a rating system is currently available only for one day and test matches, so, we cannot apply it for T-20 format. For simplicity, we have used batting average and bowling average of a player in international T-20 cricket as a measure of their performance in batting and bowling.

Now, we redefine the team selection as bi-objective optimization problem as follows:

$$\begin{aligned} & \max_{t=\{c,w,p_1\dots p_9\}} \sum_{i=c,w,p_1\dots p_9} \text{Batting Performance}(i) \\ & \max_{t=\{c,w,p_1\dots p_9\}} \sum_{i=c,w,p_1\dots p_9} \text{Bowling Performance}(i) \end{aligned}$$

The team is subject to the constraints

$$\sum_{i=c,w,p_1\dots p_9} \text{Cost}(i) < \text{Total Budget}$$

where,  $t$  represents a team comprising of  $c$ , the captain of the team,  $w$ , the wicket keeper of the team,  $p_1 \dots p_9$ , the other 9 players of the team.

After problem formulation, we run multi-objective genetic optimization over the team using the elitist non-dominated sorting genetic algorithm (NSGAI) [8]. The players are sorted according to player cost and each player is assigned a unique integer number (tag). A team is represented as a chromosome of 11 real variables with each variable denoting the tag of the players present in the team. Fitness values of each population member (i.e. team) is evaluated as the total bowling strength and total batting strength as explained in Sec. 3. Also, the maximum total budget constraint is mentioned as a constraint for the total cost of the team. Additional constraints include that no two players of the team can be same i.e. duplicates not allowed. IPL specific rules are also taken as constraints i.e. a maximum of four foreign players can be taken into the squad.

### 3 Performance Measures

#### 3.1 Batting Performance

A player's batting average is the total number of runs scored divided by the number of times he has been out [1]. Since the number of runs a player scores and how often he gets out are primarily measures of his own playing ability, and largely independent of his team mates. Thus, batting average is a good metric for an individual player's skill as a batsman. The objective function in our analysis has been taken as the summation of batting averages of all players. The problem with this approach is that some new players, even bowlers, may have batting average comparable to few of the best established batsmen due to good performance in few matches played. Hence, to avoid this scenario, the concept of primary responsibility of a player is used. A player is identified as a batsman only if he has scored at least 300 runs in international T-20

format. In calculation of team batting performance, the batting average of players identified as batsmen are only added to find net batting average. This condition is used in order to exclude batsmen who have not played enough games for their skill to be reliably assessed. So the overall batting average of team is maximized.

### 3.2 Bowling Performance

A bowler's bowling average is defined as the total number of runs conceded by the bowlers divided by the number of wickets taken by the bowler. So, the lower average is better. Again to avoid including misleading high or low averages resulting from a career spanning only a few matches, we qualify a player as bowler only if he has taken at least 20 wickets in T-20 format. Total bowling average of a team is taken as a measure of bowling performance and is minimized. Using such a strategy in optimization, results in exclusion of all bowlers from a team so that net bowling average of team goes to zero. Hence an artificial penalty bowling average for all non-bowlers needs to be added to the objective function. For our simulations, we have taken the bowling average of all non-bowlers as 100. This value is chosen to be worse than the bowler with highest bowling average.

### 3.3 Other Performance Measures

Final team selection from the trade-off front may require various other measures as explained in Sec. 5. The captaincy performance of a player is calculated as fraction of matches won out of the total number of matches played in the role of captain. It is also an important criterion in decision making process. Similarly, a player's fielding performance is measured by

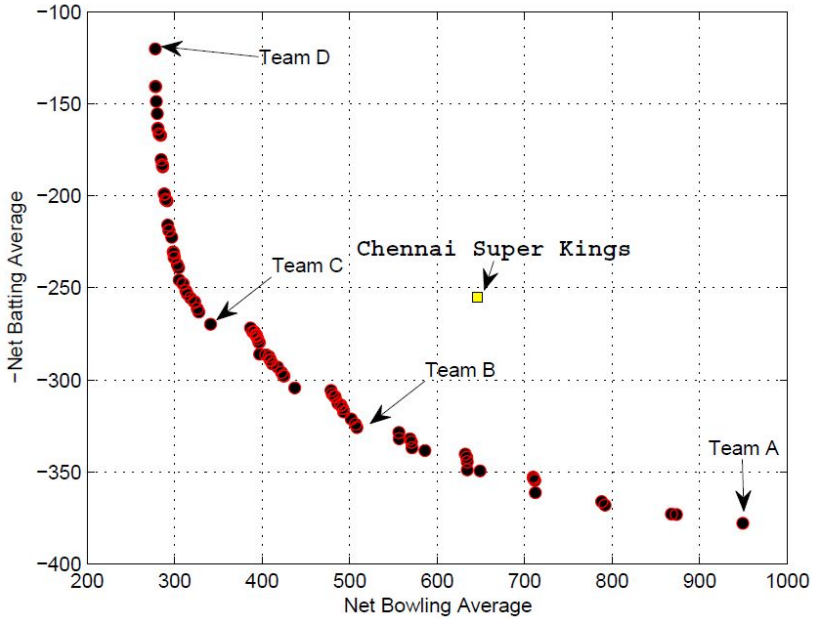
$$\text{Player's Fielding Performance} = \frac{\text{Total Catches Taken}}{\text{Total Number of Innings}}$$

Team's net fielding performance is summation of all individual players fielding performance. Number of stumping by a wicketkeeper can be taken as his wicket keeping performance measure.

## 4 Bi-objective Optimization Results

Here we present the simulation results of the above mentioned algorithms applied on our player database. The budget constraint is taken as 6 million. At least one wicketkeeper, one captain and maximum four foreign players can be included in the squad. Fig. 1 shows the Pareto-optimal front obtained. Each point on the Pareto-optimal front represents a team of 11 players. Few solution teams corresponding to points marked on the Pareto-optimal front are shown in table. The right extreme of Pareto-optimal front shows teams with very good overall batting averages while left extreme shows bowling dominant teams with low net bowling average.





**Fig. 1.** Bi-objective trade-off front. CSK team is outperformed by Teams B and C on batting and bowling performances.

To compare our results with common sense team selection methods we took the playing 11 cricketers of Chennai Super Kings (CSK), the franchise which won the finals in IPL-4. The bowling and batting fitness of the team was calculated using rules defined above along with total budget. The point representing CSK team is shown in Fig. 1. The total cost of CSK playing 11 members is estimated to be around 7.5 million. It can be seen that the team is non-optimal as well as costlier. Similar results were found for other franchise also.

#### 4.1 Budget Sensitivity Analysis

To analyze the effect of budget constraint on team's performance we have done a sensitivity analysis where the optimization process is run for a range of budget constraints and Pareto-optimal front is plotted each time. It is seen from the Fig. 2 that budget constraint affects batting dominant teams more than bowling. This is because the price difference among batsmen with high batting average and those with low average is significant. The same effect is not observed among bowlers. It also ceases to significantly affect Pareto-optimal front above a limit. Such an analysis can guide the team owners when actual prices of players are variable and the optimization is done using maximum estimated price of each player.

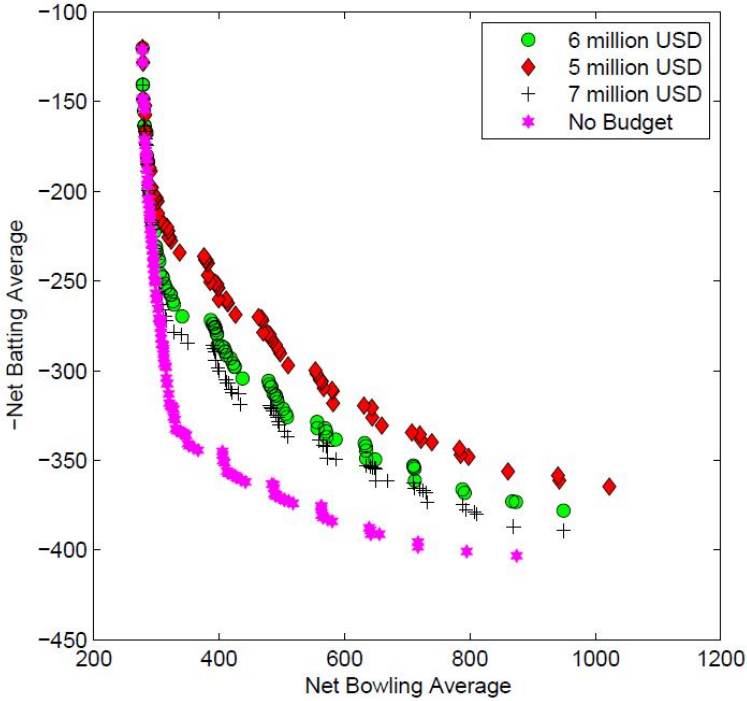


Fig. 2. Budget Sensitivity Analysis

## 5 Team Selections

The objective of the entire process is to obtain a single team of 11 members rather than a set of teams. Hence multi-criteria decision making methods need to be employed which helps in selecting any single team from the given set of mutually non-dominated teams on the trade-off front. After the initial optimization analysis, we get a Pareto-optimal front as shown in Fig. 1. Now the following method is proposed for the final team selection.

### 5.1 Knee Point Approach

Obtained trade-off front comprises of a set of points in the objective space representing various teams. For the given problem, we prefer selecting the team represented by a knee point present in the knee region of the Pareto-optimal front. Such a region is preferred because deviating from the knee region by a small change in the value of one of the objectives will compromise a large change in the value of the other objectives. Knee point can be identified by various methods [7]. Team C and Team B shown in Fig. 1 are taken as the knee points and corresponding teams are shown in Table 1.

**Table 1.** Four teams chosen from the trade-off front (Fig. 1) obtained by NSGA-II

Team A	Team B	Team C	Team D
Sachin Tendulkar	Yuvraj Singh	Yuvraj Singh	Yuvraj Singh
Michael Hussey	Nathan McCullum	JP Duminy	R Ashwin
Manoj Tiwary	Manoj Tiwary	Sudeep Tyagi	Sudeep Tyagi
Rahul Dravid	Ravindra Jadeja	Ravindra Jadeja	Nathan Rimmington
Suresh Raina	Suresh Raina	Suresh Raina	Paul Collingwood
Shaun Marsh	James Franklin	James Franklin	Steven Smith
Wriddhiman Saha	Wriddhiman Saha	Wriddhiman Saha	Wriddhiman Saha
Aaron Finch	BradHodge	Brad Hodge	Pragyan Ojha
Andrew McDonald	Andrew McDonald	Andrew McDonald	Shakib Al Hasan
Shikhar Dhawan	Shikhar Dhawan	Jaidev Unadkat	Jaidev Unadkat
Naman Ojha	Amit Mishra	Amit Mishra	Amit Mishra

The resultant team obtained shows a reasonable compromise between batting and bowling performance as compared to Team A and Team D. Since knee point is not clearly defined in the Pareto-optimal front obtained in this analysis hence we select points in the approximate knee region and apply further selection analysis to them. The knee point approach does not take into account many other factors which define a good team like fielding performance, wicketkeeper performance, expenditure, brand value of players, captain's success rate etcetera. To take into account such factors we take the solution set obtained from knee region analysis and calculate their fitness values on all such measures. New fitness value on all such factors is assigned to each team. The factors are sorted in order of importance. For example, fielding performance may be most important criteria among the other aspects mentioned above in our team selection strategy. So, we sort the solution set with respect to fielding performance and pick the solution having the best fielding side. If some other factor is also to be taken into account then we can apply in a lexicographic manner. A domination approach can also be followed. After picking a few teams from original non-dominated set with high fielding performance we shall then sort them according to other important factors, say, captaincy performance or brand value of players. Hence the team with good captain can be selected or a team with branded players can be chosen. If all preferences are exhausted and still more than one team is present in the final pool, we can get back to the expenditure criteria where the cheapest team will be preferred. Taking fielding, wicket-keeper and captain as the criteria for further analysis, the solution team obtained from knee region is mentioned below:

*Suresh Raina, Wriddhiman Saha, Yuvraj Singh, Manoj Tiwary, Roelof van der Merwe, Amit Mishra, Brad Hodge, Shikhar Dhawan, Nathan McCullum, Andrew McDonald, Ravindra Jadeja*

Using different criteria different teams can be obtained as per the requirement of attributes. The above is just an example of a good team obtained by our systematic analysis.

## 6 Conclusions

We have proposed a new methodology for objective evaluation of cricket team selection using a bi-objective genetic algorithm. An analysis of the obtained trade-off solution has been shown to result in a preferred team that has been found to have better batting and bowling averages than the winning team of the last IPL tournament. Such a methodology can be extended to include many other criteria and constraints and a better pragmatic team can be selected by the systematic procedure suggested in the paper. A standard methodology for team selection can be developed by integrating this approach with statistical analysis and using a dynamic optimization technique to be applied in auction. Abstract factors like team coordination etc. can also be used for decision making process. Nevertheless, the procedure suggested in the paper clearly demonstrates the advantage of using a bi-objective computing methodology for the cricket team selection in major league tournaments. The proposed methodology now needs some fine tuning based on other criteria for it to be used in practice.

## References

1. Cricket, <http://en.wikipedia.org/wiki/Cricket>
2. Cricket database, <http://home.iitk.ac.in/~ajindal/cricket>
3. Espn cricinfo, <http://www.espnricinfo.com/>
4. An explanation of cricket, <http://www.cs.purdue.edu/homes/hosking/cricket/explanation.htm>
5. International cricket council, <http://icc-cricket.yahoo.net>
6. Butenko, S., Gil-Lafuente, J., Pardalos, P.M.: Economics, management and optimization in sports. Springer, Heidelberg (2004)
7. Deb, K., Gupta, S.: Understanding knee points in bicriteria problems and their implications as preferred solution principles. *Engineering Optimization* 99999(1), 1–30 (2011)
8. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation* 6(2), 182–197 (2002)
9. Duarte, A., Ribeiro, C., Urrutia, S., Haeusler, E.: Referee Assignment in Sports Leagues. In: Burke, E.K., Rudová, H. (eds.) PATAT 2007. LNCS, vol. 3867, pp. 158–173. Springer, Heidelberg (2007)
10. Preston, I., Thomas, J.: Batting strategy in limited overs cricket. *Journal of the Royal Statistical Society: Series D (The Statistician)* 49(1), 95–106 (2000)
11. Régin, J.C.: Minimization of the number of breaks in sports scheduling problems using constraint programming. In: *Constraint Programming and Large Scale Discrete Optimization: DIMACS Workshop Constraint Programming and Large Scale Discrete Optimization, September 14-17. DIMACS Center, vol. 57, p. 115. Amer. Mathematical Society (2001)*
12. Swartz, T.B., Gill, P.S., Beaudoin, D., Desilva, B.M.: Optimal batting orders in one-day cricket. *Computers & Operations Research* 33(7), 1939–1950 (2006)

# Data Clustering Using Harmony Search Algorithm

Osama Moh'd Alia<sup>1</sup>, Mohammed Azmi Al-Betar<sup>2,3</sup>,  
Rajeswari Mandava<sup>2</sup>, and Ahamad Tajudin Khader<sup>2</sup>

<sup>1</sup> Faculty of Computing and Information Technology,  
University of Tabuk, Tabuk, Kingdom of Saudi Arabia  
sm\_alia@yahoo.com

<sup>2</sup> School of Computer Sciences, Universiti Sains Malaysia, 11800 Penang, Malaysia  
{mohbetar, mandava, tajudin}@cs.usm.my

<sup>3</sup> School of Computer Science, AL-Zaytoonah University of Jordan, Amman, Jordan

**Abstract.** Being one of the main challenges to clustering algorithms, the sensitivity of fuzzy c-means (FCM) and hard c-means (HCM) to tune the initial clusters centers has captured the attention of the clustering communities for quite a long time. In this study, the new evolutionary algorithm, Harmony Search (HS), is proposed as a new method aimed at addressing this problem. The proposed approach consists of two stages. In the first stage, the HS explores the search space of the given dataset to find out the near-optimal cluster centers. The cluster centers found by the HS are then evaluated using reformulated c-means objective function. In the second stage, the best cluster centers found are used as the initial cluster centers for the c-means algorithms. Our experiments show that an HS can minimize the difficulty of choosing an initialization for the c-means clustering algorithms. For purposes of evaluation, standard benchmark data are experimented with, including the Iris, BUPA liver disorders, Glass, Diabetes, etc. along with two generated data that have several local extrema.

## 1 Introduction

Clustering is a typical unsupervised learning technique for grouping similar data points according to some measure of similarity. The main goal of such technique is to minimize the inter-cluster similarity and maximize the intra-cluster similarity [1].

One of the most popular clustering algorithms is the c-means algorithm with its two types: fuzzy c-means (FCM) and hard c-means algorithm (HCM). However, selecting the initial cluster centers in these clustering algorithms is considered one of the main challenging problems. Generally, this type of clustering algorithms looks for minimizing the objective function, though it is unfortunately guaranteed only to yield local minimum [2]. Uncorrected selection of the initial cluster centers will generally lead to undesirable clustering result which will be affected by these initial values. The main cause for the local optimal problem in these algorithms is that c-means algorithms actually work similar as a hill climbing algorithm [3]. The local search-based algorithms move in one direction without performing a wider scan of the search

space. Thus the same initial cluster centers in a dataset will always generate the same cluster results; better results might as well be obtained if the algorithm is run with different initial cluster centers.

To overcome the main cause of this problem, several population-based or local search-based metaheuristic algorithms have been proposed in the last several decades including Simulating Annealing [4], Tabu Search [5], Genetic Algorithm [6,7], Particle Swarms Optimization [8], Ant Colony Algorithm [3] and Differential Evolution. The main advantages of these metaheuristic-based algorithms are their abilities to cope with local optima and effectively explore large solution spaces by maintaining, recombining and comparing several solutions simultaneously [10].

Harmony search (HS) [11] is a relatively new population-based metaheuristic optimization algorithm that imitates the music improvisation process where the musicians improvise their instruments' pitch by searching for a perfect state of harmony. It was able to attract many researchers to develop HS-based solutions for many optimization problems, see [12] [23][24][25].

The key advantage of HS lies in its ability to exploit the new suggested solution (harmony) synchronizing with exploring the search space in a parallel optimization environment. To put it in different terms, the HS works local, global, or both search strategy by means of finding an appropriate balance between exploration and exploitation through its parameter setting [13]. HS has many features that make it a preferable technique not only as standalone algorithm but also combined with other metaheuristic algorithms.

According to the latest survey on HS done in [12]; several researchers combined HS with c-means algorithms (FCM/HCM) in various domains such as heterogeneous aquifer system [14], web documents clustering [15], fuzzy classifier [16], NASA radiator clustering [17] and image clustering [18].

In this paper, a new variation of HS for solving initial centers selection problem for both HCM and FCM has been introduced. Our approach consists of two stages. In the first stage, the HS explores the search space of the given dataset to find out the near-optimal cluster centers. In the second stage, the best cluster centers found are used as the initial cluster centers for the c-means algorithms to perform clustering. The experiments conducted in this paper use standard benchmark data including the Iris, BUPA liver disorders, Glass, Diabetes, etc. along with two generated data that have several local extrema. The results of the new variation are compared with those from randomly initialized HCM/FCM.

This paper is organized as follows: Section 2 reviews the standard c-means clustering algorithms (HCM/FCM); Section 3 discusses the proposed HS-based algorithm. Experimental results are presented in Section 4, and in Section 5 a conclusion is presented.

## 2 Clustering Fundamentals

Classically, clustering algorithm is performed on a set of  $n$  patterns or objects  $X = \{x_1, x_2, \dots, x_n\}$ , each of which,  $x_i \in \mathcal{R}^d$ , is a feature vector consisting of  $d$  real-valued measurements describing the features of the object represented by  $x_i$ .

Two types of clustering algorithms are available: hard and fuzzy. In hard clustering, the goal would be to partition the dataset  $X$  into non-overlapping non-empty partitions  $G_1, \dots, G_c$ . While in fuzzy clustering algorithms the goal would be to partition the dataset  $X$  into partitions that allowed the data object to belong in a particular (possibly null) degree to every fuzzy cluster. The clustering output is a membership matrix called a partition matrix  $U = [u_{ij}]_{c \times n}$  where  $u_{ij}$  represents the membership value of the  $i$ th object to the  $j$ th cluster. The hard partition matrix is defined as:

$$M_{cn} = \left\{ U \in \mathcal{R}^{c \times n} \mid \sum_{j=1}^c u_{ij} = 1, 0 < \sum_{i=1}^n u_{ij} < n, u_{ij} \in \{0,1\}; 1 \leq j \leq c; 1 \leq i \leq n \right\}. \quad (1)$$

While the fuzzy version of this partition matrix is defined as:

$$M_{fcn} = \left\{ U \in \mathcal{R}^{c \times n} \mid \sum_{j=1}^c u_{ij} = 1, 0 < \sum_{i=1}^n u_{ij} < n, u_{ij} \in [0,1]; 1 \leq j \leq c; 1 \leq i \leq n \right\}. \quad (2)$$

Hard c-means algorithm (HCM) [19] is considered one of the most popular hard partitioning algorithms. HCM is an iterative procedure able to locally minimize the following objective function:

$$J_1(U, V) = \sum_{j=1}^c \sum_{i=1}^n U_{ij} \|x_i - v_j\|^2, \quad (3)$$

where  $\{v_j\}_{j=1}^c$  are the centers of the clusters  $c$  and  $\|\dots\|$  denotes an inner-product norm (e.g. Euclidean distance) from the data point  $x_i$  to the  $j$ th cluster center.

Fuzzy c-means algorithm (FCM) [20] is considered one of the most popular fuzzy partitioning algorithms. FCM is an iterative procedure able to locally minimize the following objective function:

$$J_m = \sum_{j=1}^c \sum_{i=1}^n U_{ij}^m \|x_i - v_j\|^2, \quad (4)$$

where  $\{v_j\}_{j=1}^c$  are the centers of the clusters  $c$  and  $\|\dots\|$  denotes an inner-product norm (e.g. Euclidean distance) from the data point  $x_i$  to the  $j$ th cluster center, and the parameter  $m \in [1, \infty)$ , is a weighting exponent on each fuzzy membership that determines the amount of fuzziness of the resulting classification and it is set to  $m = 2$ .

### 3 The Proposed Approach

Our proposed approach consists of two stages. In the first stage, the harmony search algorithm explores the search space of the given dataset to find out the near-optimal cluster centers values. In the second stage, those cluster centers with the best objective function values (i.e. minimum) are used by FCM/HCM as initial cluster centers and then the final clustering is performed. A description of these two stages is given.

### 3.1 Stage 1: Finding Near-Optimal Cluster Centers Using HS

In the following sections we describe a model of HS that represents the proposed algorithm.

#### 3.1.1 Initialization of HS Parameters

The first step of HS algorithm is to set and initialize HS parameters as follows:

1. Harmony Memory Size (HMS) (i.e. number of solution vectors in harmony memory);
2. Harmony Memory Considering Rate (HMCR), where  $HMCR \in [0,1]$  ;
3. Pitch Adjusting Rate (PAR), where  $PAR \in [0,1]$ ;
4. Stopping Criterion (i.e. number of improvisation (NI));

#### 3.1.2 Initialization of Harmony Memory

Each harmony memory (HMV) vector encodes the cluster centers of the given dataset. Each vector has a physical length of  $(c \times d)$ , where  $c$  and  $d$  are the number of clusters and the number of features, respectively. The solution vector will be as in (5):

$$HMV = \left( \overbrace{a_1 a_2 \cdots a_d}^{v_1} \overbrace{a_1 a_2 \cdots a_d}^{v_2} \cdots \overbrace{a_1 a_2 \cdots a_d}^{v_c} \right), \quad (5)$$

where  $a_i$  is a decision variable and  $a_i \in A$ .  $A$  is the set of possible range of each data features. For example, if a given dataset has 3 features with 4 classes, then the HMV could be like  $(10,30,180,30,45,201,96,140,75,1,73,13)$ , where  $(10,30,180)$  represent the cluster center values for the first class, and  $(30,45,201)$  represent the cluster center values for the second class, and so on.

In the initialization step of harmony memory (HM), each decision variable in each solution vector in HM will be initialized randomly from its data range. After that, the fitness value for each vector will be calculated by the objective function and then HM vectors will be optionally rearranged in decreasing manner.

#### 3.1.3 Improvisation of a New Harmony Vector

In each iteration of HS, a new harmony vector is generated based on the HS improvisation rules mentioned in [13]. These rules are: Memory consideration; pitch adjustment; random consideration. In HM consideration, the value of the component (i.e. decision variable) of the new vector is inherited from the possible range of the harmony memory vectors stored in HM. This is the case when a random number  $\in [0,1]$  is within the probability of HMCR; otherwise, the value of the component of the new vector is selected from the possible data range with a probability of  $(1-HMCR)$ .

Furthermore, the new vector components which are selected out of memory consideration operator are examined to be pitch adjusted with the probability of (PAR). Once a component is adjusted, its value becomes:

$$(a_i^{NEW}) = (a_i^{NEW}) \pm rand(\ ) \times bw, \quad (6)$$



where  $bw$  is an arbitrary distance bandwidth used to improve the performance of HS and  $rand(\ )$  generates a uniform random number between 0 and 1.

### 3.1.4 Update the Harmony Memory

Once the new harmony vector is generated, the fitness function is computed. Then, the new vector is compared with the worst harmony memory solution in terms of the fitness function. If it is better, the new vector is included in the harmony memory and the worst harmony is excluded.

### 3.1.5 Objective Function

In order to evaluate each HMV, we propose to use the reformulated version of standard c-means objective functions proposed in [21] as can be seen in (7) for HCM and (8) for FCM. We opted for this version of the objective function since we only use the cluster centers within the HS independent of membership matrix. Also the calculation used in the reformulated version of c-means objective functions is only use the cluster centers values independent of the membership matrix as in the standard one. Both objective functions (standard and reformulated) are equivalent but the reformulated version is less complex and less time consuming.

$$R_1 = \sum_{i=1}^n \min\{D_{1i}, D_{2i}, \dots, D_{ci}\}, \quad (7)$$

$$R_m = \sum_{i=1}^n \left( \sum_{j=1}^c D_{ji}^{\frac{1}{1-m}} \right)^{1-m}, \quad (8)$$

where  $D_j$  is  $\|x_i - v_j\|$ , the Euclidean distance from data point  $x_i$  to the  $j$ th cluster center. The minimizations of these values of  $R_1$  and  $R_m$  are the target of HS to reach the near optimal solution or meet stopping criterion.

### 3.1.6 Check the Stopping Criterion

This process is repeated until the maximum number of iterations (NI) is reached.

## 3.2 Stage 2: C-Means Clustering

Once the HS has met the stopping criterion, the solution vector from HM with best (i.e. minimum) objective function value is selected and considered as initial centers for FCM/HCM. Consequently, c-means algorithms perform data clustering in their iterative manner until the stopping criterion is met.

## 4 Experimental Results and Discussion

This section presents the results of applying HS to solve the clustering problem. The algorithm was applied to nine datasets, seven of them are real datasets, and the other two are artificial datasets.

The parameters of the HS algorithm are experimentally set as follows: HM size=10, HMCR=0.96, PAR=0.01 and the maximum number of iteration NI=50000. Furthermore, all experiments are performed on an Intel Core2Duo 2.66 GHz machine, with 2GB of RAM; while the codes are written using Matlab 2008a.

**Table 1.** Dataset descriptions

Dataset Name	Number of instances	Number of Features	Number of Classes
Artificial_1	50	1	5
Artificial_2	200	1	53
Iris	150	4	3
BUPA liver disorders	345	6	2
Glass(with 6 classes)	214	9	6
Glass(with 2 classes)	214	9	2
Haberman	306	3	2
Breast Cancer Wisconsin	569	30	2
Magic Gamma Telescope	19020	10	2
Diabetes	768	8	2

#### 4.1 Description of the Datasets

The real datasets are Iris, BUPA liver disorders, Glass, Haberman, Magic Gamma Telescope, Breast Cancer Wisconsin and Diabetes (available at: <http://archive.ics.uci.edu/ml/>). Also the Glass data set has two categories depending on the number of classes; the original dataset has six classes while the modified one has two (i.e. window glass, and non window glass classes); it was modified for simplification purposes.

Both of the artificial datasets have a single feature with multiple local extrema. The first artificial dataset, Artificial\_1, is generated from the output  $y$  of the nonlinear equation  $y = (1 + x_1^{-2} + x_2^{-1.5})^2$ , where  $1 \leq x_1, x_2 \leq 5$ . This equation is obtained from [6]. The second artificial dataset, Artificial\_2, is generated from the output  $z$  of two input nonlinear equation  $z = \sin(x)/x \times \sin(y)/y$ , where  $x, y \in [-10.5, 10.5]$ . This equation is obtained from [22]. Table 1 summarizes the main characteristics of these datasets.

#### 4.2 FCM Experiments

The quality of the solution constructed is measured in terms of the objective function and the number of iterations needed to reach an optimal solution. The experiments are designed to test the performance of HS in finding appropriate initial cluster centers for FCM algorithm compared with a standard random initialization technique used to choose cluster centers.

The results from HS initialization are marked as (HS/FCM), while the results from random initialization are marked as (RAN/FCM). Table 2 summarizes these results, where the average results from 50 trials are recorded along with standard deviation.

**Table 2.** Results from HS/FCM and RAN/FCM (bold entries indicate equal or better HS/FCM than RAN/FCM)

Dataset Name	Average $R_m$	Average $R_m$	# of iter.	# of iter.
	HS/FCM (Std.Dev.)	RAN/FCM (Std.Dev.)	HS/FCM (Std.Dev.)	RAN/FCM (Std.Dev.)
Artificial_1	<b>0.90236</b>	1.1181	<b>7</b>	16
	<b>0</b>	0.31386	<b>0</b>	5.9428
Artificial_2	<b>0.00047294</b>	0.00051345	<b>2</b>	8
	<b>0</b>	0.00012646	<b>0</b>	0.97499
Iris	<b>60.576</b>	60.576	<b>12</b>	15
	<b>0</b>	0.000018139	<b>0</b>	2.8966
BUPA liver disorders	<b>333107.6991</b>	333107.6991	<b>31</b>	31
	<b>0</b>	0.000010933	<b>0</b>	3.2001
Glass(with 6 classes)	<b>154.146</b>	154.146	45	44
	<b>0</b>	0.00003494	0	7.898
Glass(with 2 classes)	<b>556.3839</b>	556.3839	<b>11</b>	13
	<b>0</b>	8.2391E-06	<b>0</b>	0.87342
Haberman	<b>21582.6291</b>	21582.6291	<b>15</b>	16
	<b>0</b>	9.7275E-06	<b>0</b>	2.1063
Breast Cancer Wisconsin	<b>62075261</b>	62075261	<b>19</b>	22
	<b>0</b>	7.1972E-06	<b>0</b>	1.9722
Magic Gamma Telescope	<b>133807544</b>	133807544	<b>34</b>	37
	<b>0</b>	0.00001317	<b>0</b>	2.8715
Diabetes	<b>3986824.948</b>	3986824.948	<b>31</b>	34
	<b>0</b>	0.000014714	<b>0</b>	2.823

The datasets that have a single extremum as (Iris, BUPA liver disorders, Glass, Haberman, Magic Gamma Telescope, Breast Cancer Wisconsin, and Diabetes) will converge to the same extremum that each of them has. This will take place for all initialization tried in these experiments, but the speed of reaching this extremum depends on the initialization centers that are used; this will reflect on the number of iteration that FCM needs to reach the extremum as can be seen in Table 2 .

The datasets that have a multiple extrema as Artificial\_1, and Artificial\_2 will converge to a different extrema depending on the initial centers used. This will lead to different clustering results. Table 2 shows that the HS/FCM has equal or better results for all datasets with single or multiple extrema compared to the results obtained from RAN/FCM. It is also noticeable that the big improvement in the objective function results was obtained from HS/FCM in comparison with RAN/FCM when the datasets have multiple extrema.

### 4.3 HCM Experiments

The same experiments as in section 4.2 were designed for HCM, where the cluster centers initialized by HS are marked as (HS/HCM) and marked as (RAN/HCM) for random initialization. Table 3 shows these results over 50 trials where  $R_1$  value is

always better than or equal to that from randomly initialized HCM except for BUPA liver disorders dataset. The improvement in the objective function results obtained from HS/HCM in comparison with RAN/HCM is also noticeable when the datasets have multiple extrema. Table 3 also shows that the number of iterations required by HS/HCM to reach the near optimal solution is less than or equal to those obtained by RAN/HCM.

**Table 3.** Results from HS/HCM and RAN/HCM (bold entries indicate equal or better HS/HCM than RAN/HCM)

Dataset Name	Average $R_m$	Average $R_m$	# of iter.	# of iter.
	HS/HCM (Std.Dev.)	RAN/HCM (Std.Dev.)	HS/HCM (Std.Dev.)	RAN/HCM (Std.Dev.)
Artificial_1	<b>0.74969</b>	1.7759	<b>3</b>	5
	<b>0</b>	0.68725	<b>0</b>	1.9597
Artificial_2	<b>0.00048728</b>	0.051385	<b>2</b>	4
	<b>0</b>	0.027815	<b>0</b>	1.2856
Iris	<b>78.9408</b>	88.0479	<b>4</b>	7
	<b>0</b>	22.8009	<b>0</b>	2.9031
BUPA liver disorders	424059.2166	424057.6499	<b>9</b>	10
	0	11.0779	<b>0</b>	2.0102
Glass(with 6 classes)	<b>338.7449</b>	412.8961	<b>5</b>	11
	<b>0</b>	86.6537	<b>0</b>	4.1145
Glass(with 2 classes)	<b>819.6293</b>	852.4213	<b>4</b>	6
	<b>0</b>	101.4054	<b>0</b>	2.0127
Haberman	<b>30507.2736</b>	31330.8083	10	8
	<b>0</b>	3095.7777	0	3.2059
Breast Cancer Wisconsin	<b>77943099.88</b>	77943099.88	<b>5</b>	7
	<b>0</b>	4.516E-08	<b>0</b>	1.7023
Magic Gamma Telescope	<b>200963098.1</b>	200963098.1	28	24
	<b>0</b>	1.505E-07	0	7.2002
Diabetes	<b>5142376.456</b>	5142376.456	16	14
	<b>0</b>	0	0	2.6033

## 5 Conclusion and Future Work

In this paper we present the Harmony Search optimization algorithm to overcome cluster centers initialization problem in clustering algorithms (FCM/ HCM). This step is important in data clustering since the incorrect initialization of cluster centers will lead to a faulty clustering process. Harmony search algorithm works globally and locally in the search space to find the appropriate cluster centers. The experiment evaluation shows that the algorithm can tackle this problem intelligently; the advantages of this algorithm over standard FCM/HCM are clearer in datasets with multiple extremes. More extensive comparison studies between the optimization algorithms that solve clustering algorithm problems are needed.

**Acknowledgments.** The second author is grateful to be awarded a Postdoctoral Fellowship from the school of Computer Sciences (USM).

## References

1. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. *ACM Comput. Surv.* 31(3), 264–323 (1999)
2. Hathaway, R.J., Bezdek, J.C.: Local convergence of the fuzzy c-means algorithms. *Pattern Recognition* 19(6), 477–480 (1986)
3. Kanade, P.M., Hall, L.O.: Fuzzy ants and clustering. *IEEE Transactions on Systems, Man and Cybernetics, Part A* 37(5), 758–769 (2007)
4. Selim, S.Z., Alsultan, K.: A simulated annealing algorithm for the clustering problem. *Pattern Recognition* 24(10), 1003–1008 (1991)
5. Al-Sultan, K.S.: A tabu search approach to the clustering problem. *Pattern Recognition* 28(9), 1443–1451 (1995)
6. Hall, L.O., Ozyurt, I.B., Bezdek, J.C.: Clustering with a genetically optimized approach. *IEEE Transactions on Evolutionary Computation* 3(2), 103–112 (1999)
7. Bandyopadhyay, S., Maulik, U.: An evolutionary technique based on k-means algorithm for optimal clustering in rn. *Information Sciences* 146(1-4), 221–237 (2002)
8. Lili, L., Xiyu, L., Mingming, X.: A novel fuzzy clustering based on particle swarm optimization. In: *First IEEE International Symposium on Information Technologies and Applications in Education, ISITAE*, pp. 88–90 (2007)
9. Maulik, U., Saha, I.: Modified differential evolution based fuzzy clustering for pixel classification in remote sensing imagery. *Pattern Recognition* 42(9), 2135–2149 (2009)
10. Paterlini, S., Krink, T.: Differential evolution and particle swarm optimisation in partitional clustering. *Computational Statistics & Data Analysis* 50(5), 1220–1247 (2006)
11. Geem, Z.W., Kim, J.H., Loganathan, G.: A new heuristic optimization algorithm: harmony search. *Simulation* 76(2), 60–68 (2001)
12. Alia, O., Mandava, R.: The variants of the harmony search algorithm: an overview. *Artificial Intelligence Review* 36, 49–68 (2011), 10.1007/s10462-010-9201-y
13. Geem, Z.W.: *Music-inspired Harmony Search Algorithm Theory and Applications*. Springer, Heidelberg (2009)
14. Ayvaz, M.T.: Simultaneous determination of aquifer parameters and zone structures with fuzzy c-means clustering and meta-heuristic harmony search algorithm. *Advances in Water Resources* 30(11), 2326–2338 (2007)
15. Mahdavi, M., Chehreghani, M.H., Abolhassani, H., Forsati, R.: Novel meta-heuristic algorithms for clustering web documents. *Applied Mathematics and Computation* 201(1-2), 441–451 (2008)
16. Wang, X., Gao, X.Z., Ovaska, S.J.: A hybrid optimization method for fuzzy classification systems. In: *Eighth International Conference on Hybrid Intelligent Systems, HIS 2008*, pp. 264–271 (2008)
17. Malaki, M., Pourbagheri, J.A., Abolhassani, H.: A combinatory approach to fuzzy clustering with harmony search and its applications to space shuttle data. In: *SCIS & ISIS 2008*, Nagoya, Japan (2008)
18. Alia, O.M., Mandava, R., Aziz, M.E.: A hybrid harmony search algorithm to MRI brain segmentation. In: *The 9th IEEE International Conference on Cognitive Informatics, ICCI 2010*, pp. 712–719. IEEE, Tsinghua University (2010)

19. Forgy, E.: Cluster analysis of multivariate data: Efficiency vs. interpretability of classifications. *Biometrics* 21(3), 768 (1965)
20. Bezdek, J.C.: *Pattern Recognition with Fuzzy Objective Function Algorithms*. Kluwer Academic Publishers (1981)
21. Hathaway, R.J., Bezdek, J.C.: Optimization of clustering criteria by reformulation. *IEEE Transactions on Fuzzy Systems* 3(2), 241–245 (1995)
22. Alata, M., Molhim, M., Ramini, A.: Optimizing of fuzzy c-means clustering algorithm using GA. *Proceedings of World Academy of Science, Engineering and Technology* 29 (2008)
23. Al-Betar, M., Khader, A.: A hybrid harmony search for university course timetabling. In: *Proceedings of the 4nd Multidisciplinary Conference on scheduling: Theory and Applications (MISTA 2009)*, Dublin, Ireland, pp. 10–12 (August 2009)
24. Al-Betar, M., Khader, A.: A harmony search algorithm for university course timetabling. *Annals of Operations Research*, 1–29 (2008)
25. Al-Betar, M., Khader, A., Nadi, F.: Selection mechanisms in memory consideration for examination timetabling with harmony search. In: *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, pp. 1203–1210. ACM (2010)

# Application of Swarm Intelligence to a Two-Fold Optimization Scheme for Trajectory Planning of a Robot Arm

Tathagata Chakraborti<sup>1</sup>, Abhronil Sengupta<sup>1</sup>, Amit Konar<sup>1</sup>, and Ramadoss Janarthanan<sup>2</sup>

<sup>1</sup> Dept. of Electronics and Telecommunication Engg., Jadavpur University, Kolkata, India  
tathagata.net@live.com, senguptaabhronil@gmail.com,  
konaramit@yahoo.co.in

<sup>2</sup> Department IT, Jaya Engineering College, Chennai, India  
srmjana\_73@yahoo.com

**Abstract.** Motion planning of a robotic arm has been an important area of research for the last decade with the growing application of robot arms in medical science and industries. In this paper the problem of motion planning has been dealt with in two stages, first by developing appropriate cost functions to determine a set of via points and then fitting an optimal energy trajectory. Lbest Particle Swarm Optimization has been used to solve the minimization problem and its relative performance with respect to two other popular evolutionary algorithms, Differential Evolution and Invasive Weed Optimization, has been studied. Experiments indicate swarm intelligence techniques to be far more efficient to solve the optimization problem.

## 1 Introduction

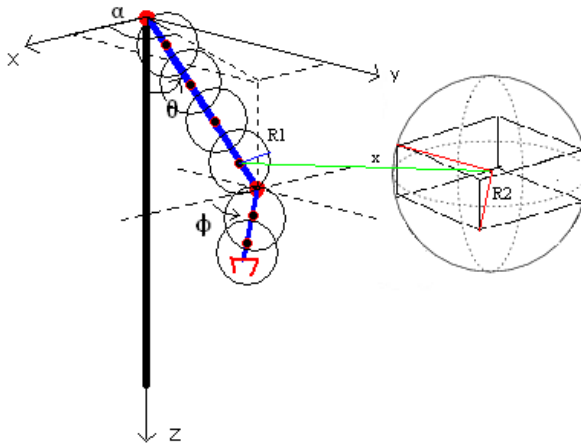
Motion planning of a robot arm requires the formulation of a specific set of via points through which the end effector must pass in order to avoid collision with obstacles in its environment while moving from an initial arm configuration to a final arm configuration. The next step requires the determination of smooth trajectory fitting through the obtained via points. However, the determination of optimal trajectory is the main area of concern.

Several research works have been conducted in this field during the past few years. Saab and VanPutte [1] used topographical maps to solve the problem. In [2] an algorithm has been proposed for obstacle avoidance using convexity of arm links and obstacles. Ziqiang Mao and T.C. Hsia et al. [3] employed neural networks to solve the inverse kinematics problem of redundant robot arms. Recently genetic algorithms (GA) have been used in this field. Traces of work by this method are found in [4-6]. Other researches in this field include Potential Field method [7].

In our paper we propose to solve the motion planning problem by developing an appropriate fitness function. In the first step a cost function has been developed to determine a set of via points subject to obstacle avoidance and other motion constraints. In the next step another cost function has been formulated to determine the trajectory joining the obtained via points by minimizing mechanical energy consumption.

Classical optimization techniques are not applicable here because of the roughness of the objective function surface. We therefore use derivative free optimization algorithms for this purpose. The first one used is Lbest PSO. The basic PSO algorithm is based on the sociological flocking behavior of birds. The Lbest PSO model is a variant of the basic PSO algorithm where each particle interacts directly with other particles of its local neighborhood [10]. The second optimization technique used for analysis is Differential Evolution (DE) which is guided by the Evolution of Chromosomes [9]. The third algorithm used is Invasive Weed Optimization (IWO) proposed by Mehrabian and Lucas [8]. It mimics the colonizing behavior of weeds.

In this paper we present a comparative analysis of the efficiency of these evolutionary algorithms in finding the optimal solution. The promising results displayed by Lbest PSO algorithm may open up new avenues of application of swarm intelligence in control of robotic systems.



**Fig. 1.** Model of the Robot Arm

## 2 Description of the Robot Arm

In the following discussion we use a robot arm with two links and two movable sections each having one degree of freedom. Their movements are described by angles theta and phi which are defined with respect to the co-ordinate system as shown in Figure 1. The first section of length  $L_1$  moves in the vertical plane as described by theta ( $\theta$ ) measured from the positive z-axis. The vertical plane in which it moves is displaced from the x-axis by an angle  $\alpha$ . The second section of length  $L_2$  moves in the horizontal plane as described by phi ( $\phi$ ) measured from the positive x-axis.



### 3 Lbest Particle Swarm Optimization

An initial population of particles (candidate solutions) is generated randomly over the D dimensional space. Each particle has the following characteristics associated with it:  $\mathbf{x}_i(t)$  which represents the present location of particle and  $\mathbf{v}_i(t)$  which represents the present velocity of particle. Local neighborhoods each of size d are formed at each generation by grouping together particles with minimum Euclidean distance between them.

The individual best fitness and the corresponding location are updated in this step. The best position for each local neighborhood is also updated.

The equation for velocity update of the  $i^{\text{th}}$  particle is given by equation (1).  $\omega$  is the inertial coefficient. Here  $r_1$  and  $r_2$  are two independent random numbers where  $r_1 \sim U(0,1)$  and  $r_2 \sim U(0,1)$ . The values of  $r_1$  and  $r_2$  are multiplied by scaling factors  $c_1$  and  $c_2$  where  $0 < c_1 < 2$  and  $0 < c_2 < 2$ .  $\mathbf{p}_i(t)$  represents personal best position of particle at time t and  $\mathbf{l}_i(t)$  represents the best position that the local neighborhood has encountered so far.

$$\vec{v}_i(t+1) = \omega \cdot \vec{v}_i(t) + c_1 \cdot r_1 \cdot (\vec{p}_i(t) - \vec{x}_i(t)) + c_2 \cdot r_2 \cdot (\vec{l}_i(t) - \vec{x}_i(t)). \quad (1)$$

In order to ensure that the particle remains bounded in the search space, the velocity of the particle is clamped to  $[-v_{\max}, v_{\max}]$ . The maximum velocity is usually chosen as  $v_{\max} = k \cdot x_{\max}$  where the search space is defined in the range  $[-x_{\max}, x_{\max}]$ .

Next the position of the particle is updated according to the equation:

$$\vec{x}_i(t+1) = \vec{x}_i(t) + \vec{v}_i(t+1). \quad (2)$$

This process is repeated until maximum number of iterations is reached.

## 4 Formulation of Cost Function for via Point Determination

### 4.1 Minimization of Redundant Joint Rotations

Here we purpose to minimize the energy consumed in redundant arm movements. Thus the problem reduces to one of ensuring that the final angle is attained as quickly as possible. Thus the cost function becomes

$$f = K1 \cdot |\theta_f - \theta| + K2 \cdot |\phi_f - \phi|. \quad (3)$$

where K1, K2 are constants of proportionality and  $(\theta_f, \phi_f)$  denotes the goal position. In the above equation  $\theta = \theta_{prev} + \Delta\theta$  and  $\phi = \phi_{prev} + \Delta\phi$  where  $(\theta_{prev}, \phi_{prev})$  represents the previous via point and  $(\Delta\theta, \Delta\phi)$  denotes the angular displacement between the two via points.

### 4.2 Obstacle Avoidance

We model the robot arm as a series of consecutive spheres of radius R1, where R1 is determined by the amount of safety margin required. Now the obstacle may be of

different shapes and sizes and developing different penalty terms for each of them is a rather futile process. As a simple model we have approximated the obstacles in the robot environment by equivalent circumspheres.

If the distance 'x' between the centre of the sphere on the arm and the centre of the sphere representing the obstacle is less than  $R1 + R2$  then there is a chance of collision and hence in such cases the cost function should incorporate a large penalty term. In cases where this distance is larger, the penalty term should be negligible. Thus the penalty component of the cost function finally takes the form (for the  $i^{\text{th}}$  sphere and  $j^{\text{th}}$  obstacle):

$$\sum_i \sum_j K3. C_{ij}. \exp(-x_{ij}/(R1_i + R2_j)). \quad (4)$$

where K3 is a constant of proportionality and is in general much greater than K1 and K2 since this lends more weight to the penalty term as obstacles must be avoided at any cost. Here,  $C_{ij} = 1$  if  $x_{ij} < R1_i + R2_j$  and is equal to zero otherwise.

Thus in its final form, the cost function becomes:

$$f = K1. |\theta_f - \theta| + K2. |\phi_f - \phi| + \sum_i \sum_j K3. C_{ij}. \exp(-x_{ij}/(R1_i + R2_j)). \quad (5)$$

To ensure uniform distribution of via-points in the joint space, the optimizer has been employed to produce optimized values of  $\Delta\theta$  and  $\Delta\phi$  within a certain given range ( $-\Delta\theta_{\max}$ ,  $\Delta\theta_{\max}$ ) and ( $-\Delta\phi_{\max}$ ,  $\Delta\phi_{\max}$ ) which have been added to the  $\theta_{\text{prev}}$  and  $\phi_{\text{prev}}$  values, and the process is repeated till the goal is reached.

## 5 Formulation of Cost Function for Trajectory Planning

Here we develop an energy efficient method of fitting a smooth trajectory to the set of via-points found above. Let us consider that we have  $n+1$  via-points (including initial and final points). We fit smooth cubic polynomials for theta and phi as functions of time in between each of these points as shown below:

$$\theta = a_0 + a_1 t + a_2 t^2 + a_3 t^3. \quad (6)$$

$$\phi = b_0 + b_1 t + b_2 t^2 + b_3 t^3. \quad (7)$$

where the coefficients are determined partly by a set of boundary conditions and partly by energy minimization criterion. The boundary conditions for the polynomial between the  $i^{\text{th}}$  and  $(i+1)^{\text{th}}$  via-points are as follows:

$$\theta = \theta_i, \phi = \phi_i; \quad \bar{\theta} = \bar{\theta}_i, \bar{\phi} = \bar{\phi}_i \quad \text{at } t = 0. \quad (8)$$

$$\theta = \theta_{i+1}, \phi = \phi_{i+1} \quad \text{at } t = T. \quad (9)$$

where  $i=1, 2, \dots, n$  and  $T$  is the time to move from the  $i^{\text{th}}$  to the  $(i+1)^{\text{th}}$  via-point, and  $\bar{\theta}, \bar{\phi}$  are the first time derivatives of theta and phi respectively.

Evidently the above conditions give three equations for four unknown coefficients, and the final equation is provided by the energy term. At any point the mechanical energy of the arm will be given by the summation of kinetic and potential energies of each of the arm sections. The total mechanical energy of the system is equal to

$$E = \frac{1}{2} m_2 L_1^2 \bar{\theta}^2 + \frac{1}{6} (m_2 L_1^2 \bar{\theta}^2 + m_2 L_2^2 \bar{\phi}^2) + \frac{1}{2} m_2 L_1 L_2 \cos(\theta) \cos(\phi) \bar{\theta} \bar{\phi} + gH(m_1 + m_2) - gL_1 \left( \frac{1}{2} m_1 + m_2 \right) \cos(\theta). \quad (10)$$

In this equation we put the expressions for  $\theta$ ,  $\phi$  and  $\bar{\theta}, \bar{\phi}$ ; and replace  $a_2, b_2$  in terms of  $a_3, b_3$  so that  $E$  is now a function of time, and  $a_3, b_3$ . The energy integrated over a single time interval will give a measure of the total energy consumed and this is a function of  $a_3$  and  $b_3$  (the integration is done using recursive adaptive Simpson quadrature technique). Thus we can find an optimum value of the remaining coefficient by minimizing  $F_k$ , where

$$F_k(a_3, b_3) = \int_0^T E. dt. \quad (11)$$

#### **MOTION PLANNING ALGORITHM**

**INPUT:** Initial and Final Arm Positions

**OUTPUT:** Energy Efficient Trajectory

**Begin**

**Step1:** Determination of Via Points

**Repeat**

**Run** Lbest PSO

Input: cost function  $f$ , optimizer parameters;

Output: optimized values of  $(\Delta\theta, \Delta\phi)$ ;

Update  $\theta_{\text{prev}}$  and  $\phi_{\text{prev}}$  in  $f$ ;

**If** goal is reached,

**Stop**;

**Else Continue**;

Obtain  $n + 1$  via points;

**Step2:** Determination of Optimal Energy Trajectory

**For**  $i = 1$  to  $n$

**Run** Lbest PSO

Input: cost function  $F_k$ , optimizer parameters;

Output: optimized values of  $a_3, b_3$ ;

**End For**

**End**

## **6 Experimental Results**

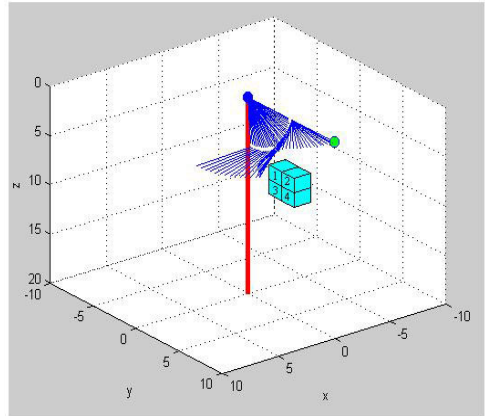
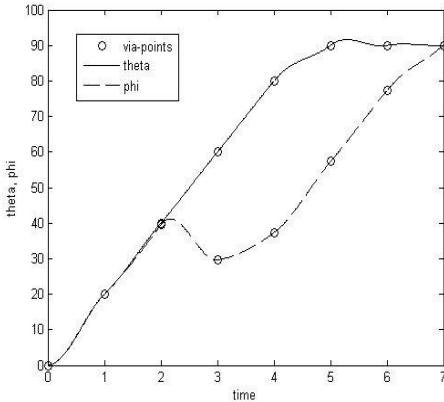
### **6.1 Optimized Trajectory Generation**

The robot arm starts from initial position defined by  $(\theta, \phi)_{\text{initial}}$  and reaches the goal defined by  $(\theta, \phi)_{\text{final}}$ . We have considered cubical obstacles with edge length 1.5, so

that R2 for the equivalent sphere becomes  $(\sqrt{3}/2)*1.5=1.29$ . The path taken by the robot arm is shown below in Figure 2. The optimal set of parameters required to generate the trajectory was determined by optimizing the cost functions with each of the 3 algorithms over a series of 10 runs and then averaging the result.

**Table 1.** Simulation Parameters

Simulation parameters	Estimated values	Simulation parameters	Estimated values
$(\theta, \phi)_{initial}$	(0,0) deg	m1	1
$(\theta, \phi)_{final}$	(90,90) deg	m2	1
K1	10	L1	5
K2	10	L2	5
K3	100	obstacle1	(3,8,3)
K4	10000	obstacle2	(3,9,5,3)
R1	0.5	obstacle3	(3,8,4,5)
R2	1.29	obstacle4	(3,9,5,4,5)
H	20	$(-\Delta\theta, \Delta\theta)$	$(-20,20)$ deg
T	1	$(-\Delta\phi, \Delta\phi)$	$(-20,20)$ deg



**Fig. 2.** Joint space trajectory plot

### 6.2 Optimizer Parameters

For the performance analysis of the three algorithms we use 10 particles to scourge the problem space. The parameters for the various optimization algorithms are described below.

**Lbest PSO.** Local neighborhoods each of size  $d=5$  are formed at each generation by grouping together particles with minimum Euclidean distance between them. The

inertial coefficient has been made to decrease linearly over the iterations from a value of 0.9 to 0.4. The scaling factors for the social and cognitive components in the velocity update equation have been chosen to be equal to 2. The maximum velocity for each particle has been set equal to  $x_{max}$ .

**IWO.** The initial weed population has been chosen equal to 5. The maximum and minimum seed counts for each generation have been set equal to 5 and 0 respectively. The modulation index is set to 3. The initial and final values of the standard deviations have been taken to be equal to 10% and 0.004% of the search range respectively.

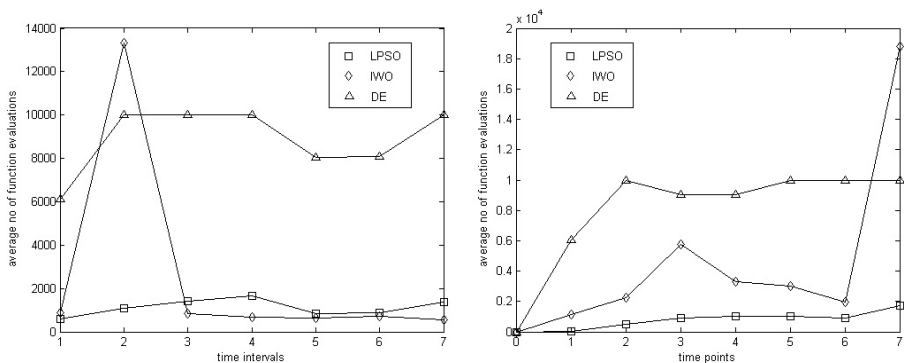
**DE.** The scaling factor used for the generation of donor chromosome is 0.8. After mutation, recombination takes place. The Crossover Constant used for the generation of the trial offspring vector was taken to be equal to 0.96.

### 6.3 Performance Evaluation

Each of the algorithms was run 10 times and the average number of fitness function evaluations required to attain the optimal set of parameters (within specified error limits) were evaluated. The following graphs illustrate the performance of the algorithms to track the optimal set of via points and trajectory respectively.

**Table 2.** Total Number of Fitness Function Evaluations Throughout Entire Journey

Description	Lbest PSO	DE	IWO
Via-Point Determination	<b>6099</b>	64142	36159
Trajectory Generation	<b>7850</b>	62228	17632



**Fig. 3.** Comparison of optimizer performances

## 7 Conclusions

The algorithm determines a set of potential via points for obstacle avoidance. The obtained trajectory ensures smooth motion of the end-effector and optimizes energy expenditure during motion from one via point to the next. The simulation results clearly indicate that swarm intelligence is a far better approach for optimization in this scenario and may be utilized as an effective optimization tool for evolutionary robotics.

## References

1. Saab, Y., VanPutte, M.: Shortest Path Planning on Topographical Maps. *IEEE Transactions on Systems, Man, and Cybernetics–Part A* 29(1), 139–150 (1999)
2. Gilbert, E.G., Johnson, D.E.: Distance Function and Their Application to Robot Path Planning in the Presence of Obstacles. *IEEE J. of Robotics and Automation* RA-1(1) (1985)
3. Tian, L., Collins, C.: An Effective Robot Trajectory Planning Using a Genetic Algorithm. *Elsevier, Mechatronics* 14, 455–470 (2004)
4. Zalzal, A.M.S., Chan, K.K.: An Evolutionary Solution for the Control of Mechanical Arms. In: *Proceedings of ICARCV 1994, Singapore* (1994)
5. Pack, D., Toussaint, G., Haupt, R.: Robot Trajectory Planning Using a Genetic Algorithm. In: *SPIE 1996, vol. 2824, pp. 171–182* (1996)
6. Mao, Z., Hsia, T.C.: Obstacle Avoidance Inverse Kinematics Solution of Redundant Robots by Neural Networks. *Robotica* 15, 3–10 (1997)
7. Khatib, O.: Real-time Obstacle Avoidance for Manipulators and Mobile Manipulators. *Int. J. of Rob. Res.* 5(1), 90–98 (1986)
8. Mehrabian, A.R., Lucas, C.: A Novel Numerical Optimization Algorithm Inspired from Weed Colonization. *Ecological Informatics* 1, 355–366 (2006)
9. Konar, A., Das, S.: Recent Advances in Evolutionary Search and Optimization Algorithms. In: *NGMS 2006, BESU, Shibpur, Howrah, India, January 11-13* (1996)
10. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In: *Proceedings of IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948* (1995)

# Two Hybrid Meta-heuristic Approaches for Minimum Dominating Set Problem

Anupama Potluri and Alok Singh

Department of Computer and Information Sciences  
University of Hyderabad, Hyderabad 500046, India  
{apcs, alokcs}@uohyd.ernet.in

**Abstract.** Minimum dominating set, which is an NP-hard problem, finds many practical uses in diverse domains. A greedy algorithm to compute the minimum dominating set is proven to be the optimal approximate algorithm unless  $P = NP$ . Meta-heuristics, generally, find solutions better than simple greedy approximate algorithms as they explore the search space better without incurring the cost of an exponential algorithm. However, there are hardly any studies of application of meta-heuristic techniques for this problem. In some applications it is important to minimize the dominating set as much as possible to reduce cost and/or time to perform other operations based on the dominating set. In this paper, we propose a hybrid genetic algorithm and an ant-colony optimization (ACO) algorithm enhanced with local search. We compare the performance of these two hybrid algorithms against the solutions obtained using the greedy heuristic and another hybrid genetic algorithm proposed in literature. We find that the ACO algorithm enhanced with a minimization heuristic performs better than all other algorithms in almost all instances.

**Keywords:** Ant-Colony Optimization, Genetic Algorithm, Heuristic, Minimum Dominating Set.

## 1 Introduction

A dominating set (DS) of a graph  $G = (V, E)$  is a subset  $S \subseteq V$  such that every node  $v \in V$  is either a member of  $S$  or is adjacent to a member of  $S$ . A dominating set with minimum cardinality is called as the Minimum Dominating Set (MDS). This is proven to be an NP-hard problem [8]. Minimum dominating sets are extensively used in wireless networks for clustering and formation of backbone used for routing. In addition, they are also used in information retrieval, facility location and so on. Reducing the cardinality of a DS is useful in reducing the cost of locating facilities. In information retrieval, a query is matched against the dominating nodes. If the cardinality of the dominating set is reduced, the time for retrieving information is accordingly reduced.

The greedy heuristic [13] for finding a minimum dominating set (MDS) is based on that given for the set cover by Chvatal [2]. This returns a solution at most  $(\ln \Delta \times Opt)$  where  $\Delta$  is the maximum degree of a node in the graph. It is proven to be the optimal approximate solution unless  $P = NP$  [13]. In this algorithm, the nodes are all initially colored WHITE. A node which is in the dominating set is colored BLACK and all

dominated nodes are colored GREY. The algorithm works by selecting a non-BLACK node with the maximum white degree and including it in the dominating set. The node is then colored BLACK and all its neighbors are colored GREY. This is repeated until all nodes are dominated, i.e., there are no more WHITE nodes in the graph. A polynomial-time approximation scheme (PTAS) for the computation of MDS is presented in [9], but this is applicable only for polynomially bounded growth graphs such as unit disk graphs. A polynomially-bounded growth graph is a graph where the number of independent nodes in the  $r$ -hop neighborhood of a node are bounded by a polynomial  $f(r) = O(r^c)$  for some constant  $c \geq 1$ . However, the PTAS algorithm is not practical for unit disk graphs (UDG) with higher degree of connectivity or large graphs. We observed that the algorithm takes a lot of time when we implemented the PTAS algorithm for Minimum Independent Dominating Set problem.

Recently, the problem of MDS has been solved using a hybrid genetic algorithm [4]. In this paper, the authors use a generational GA with a linear rank selection algorithm to select parents for the standard one-point crossover and uniform mutation to generate members of the new population. It uses three procedures to reduce the cardinality of the solution computed. One is called *Filtering* which basically checks if a node can be removed from the dominating set without affecting the domination property. Procedure *Local Search* tries to randomly add nodes to the dominating set proportional to its degree if the generated population member is not a DS; it deletes a node randomly in inverse proportion to its degree if it is a DS. In both cases, if the fitness value increases, it retains the change to the chromosome. The final procedure in the paper, *EliteInspiration* constructs the intersection of the three best DS solutions found so far and then tries to minimize the cardinality. However, in our experimentation, we found that in some cases, this algorithm does not even find a DS. We modified the algorithm such that it always returns a DS as output. We changed it as follows: after finding the intersection of the best  $n_{core}$  members of the DS population, if the resultant solution is not a DS, we keep adding nodes to it using the greedy heuristic until the member is a DS. We then, applied *Filtering* to reduce the cardinality of this solution. We found that with these changes, the solution returned by their algorithm is better than a straightforward implementation of their paper. The results presented in this paper are with these changes incorporated.

In this paper, we propose a hybrid genetic algorithm which is vastly different from that in [4] and an Ant-Colony Optimization (ACO) algorithm that uses local search to minimize the DS found by each ant in every iteration. We compared the performance of these two meta-heuristic algorithms proposed here with those in [4] and the greedy heuristic. We experimented using unit disk graphs and other general graph instances. We found our algorithms to be consistently better than the greedy heuristic and the HGA in [4]. In fact, we found that the heuristic is better than the HGA of [4]. We find that the Ant-Colony Optimization (ACO) algorithm combined with a local search heuristic performs better than all the other algorithms studied.

The rest of the paper is organized as follows: we present the steady-state genetic algorithm with the minimization heuristic in section 2 and the ant-colony optimization algorithm in section 3. The comparison of the results between the heuristic, hybrid genetic algorithm in [4] and our proposed algorithms is given in section 4. We end the paper with the conclusions in 5.



---

**Algorithm 1. Hybrid Genetic Algorithm for MDS**


---

```

Generate Initial Population,  $I$ 
 $F :=$  fitness of best member of  $I$ 
 $b :=$  Best member of  $I$ 
while  $gen < MAXGEN$  do
  if ( $p < p_e$ ) then
    Select parents  $p_1$  and  $p_2$  using binary tournament selection
     $C :=$  crossover( $p_1, p_2$ )
     $C :=$  mutate( $C$ )
  else
    Generate  $C$  randomly
  end if
  if ( $p < p_h$ ) then
     $C :=$  HeuristicRepair( $C$ )
  else
     $C :=$  RandomRepair( $C$ )
  end if
   $C :=$  Minimize( $C$ )
  if unique( $C$ ) then
    Replace worst member of the population with  $C$ 
    if  $f(C) < F$  then
       $F := f(C)$ 
       $b := C$ 
    end if
     $gen := gen + 1$ 
  end if
end while
return  $b$ 

```

---

## 2 The Hybrid Genetic Algorithm

We have used a steady-state genetic algorithm [3] to solve the MDS problem. The chromosome is represented as a bit vector of size  $N$  where  $N$  is the number of nodes in the given graph. The fitness of the solution is the cardinality of the generated dominating set (DS). We start off with an initial population of 100 members. Each member is generated by randomly setting bits in the bit vector with a probability of 0.3. If the generated solution is not a DS, we repair it using the greedy heuristic with probability  $p_h$  or by adding nodes randomly. Then, we minimize the cardinality as follows: if any node of the DS and all its neighbors are covered by other nodes in the DS, it is redundant. Such a node can be removed from the DS, thus reducing cardinality without affecting domination property. We repeatedly remove redundant nodes until there are no more such nodes in the computed DS. In each iteration, we remove a redundant node randomly with probability  $p_r$  or using the policy of lowest degree redundant node with probability  $(1 - p_r)$ . If the solution thus generated is unique, it is added to the population. For creating a new child, binary tournament is used to select two parents for crossover. The parent with a better fitness is selected with probability  $p_{better}$ . We use the crossover method proposed

by Beasley and Chu [11] to create the child member. Let the parents be  $p_1$  and  $p_2$  and their respective fitness values  $f(p_1)$  and  $f(p_2)$ . The bits in the child are inherited from parent  $p_1$  with probability  $\frac{f(p_2)}{f(p_1)+f(p_2)}$  and from parent  $p_2$  with probability  $\frac{f(p_1)}{f(p_1)+f(p_2)}$ . This method ensures that bits are more often inherited from a parent with a better fitness ensuring that the fitness of the child is likely to be better. Crossover is not applied always but with probability  $p_c$ ; in other instances, a new member is generated randomly to diversify the population. A simple bit flip mutation scheme is used with probability  $p_m$ . If the new child generated is not a DS, the greedy heuristic is used to add nodes until it is a DS. We, then, minimize the solution as specified in initial population generation. We replace the worst member of the previous generation with this new member. Algorithm 1 provides the pseudo-code of our approach.

### 3 The ACO Local Search Algorithm

First of all, we experimented with a standard ACO with no enhancements. We found the results to be much worse than even the greedy heuristic. We tried enhancing the algorithm by calculating probability based on both the pheromone value as well as the degree of a node as proposed for the minimum weighted vertex cover problem by Shyu et al [10]. In this, the higher the degree of a node, the higher the probability with which it is selected. However, we found that this was not generating a good solution either. We then proposed a local search similar to that used in the maximum clique problem in [12] as an enhancement to the standard ACO. We found this to be the most effective algorithm and this is what is described here.

---

#### Algorithm 2. ACO Algorithm for Computing MDS

---

```

F := N
b :=  $\phi$ 
for I := 1  $\rightarrow$  MAX - ITER do
  for A := 1  $\rightarrow$  MAX - ANTS do
     $p_i := \frac{\tau_i}{\sum_j \tau_j}$ 
    Add node i to S with probability  $p_i$  until S is a dominating set
    S := Minimize(S)
  end for
  D := Best(S)
  if  $f(D) < F$  then
    F :=  $f(D)$ 
    b := D
  end if
  Update_Pheromone(D)
end for
return b

```

---

In our hybrid ACO algorithm, in each cycle, a total of  $N_{ants}$  perform a random walk of the graph until they discover a DS. Initially, we deposit a pheromone value of  $\tau_0 = 10.0$  on each node. Each ant chooses the next candidate to include in the DS based on the probability of the node which is calculated as  $p_i = \frac{\tau_i}{\sum_j \tau_j}$ , where  $\tau_i$  is the pheromone concentration on node  $i$ . After the random walk of an ant is completed, the DS found is minimized using the concept of redundant nodes as described in the previous section. At the end of a cycle, we use the best ant found in that cycle to reinforce the pheromone value. The pheromone concentration on the nodes of the best ant found is updated using the formula  $\tau_i = \rho \times \tau_i + \frac{1}{10+f-F}$  where  $f$  is the fitness of the best ant in this cycle and  $F$  is the fitness of the best ant found so far and  $\rho$  is the pheromone persistence rate. This is similar to the formula used to update pheromone value in [11]. For all the other nodes in the graph, the pheromone is evaporated using the formula  $\tau_i = \rho \times \tau_i$ ; if the resultant value is less than  $\tau_{min}$ , the value is set to  $\tau_{min}$ . We run the algorithm for a total of specified cycles. We present the results using only the local search in this paper due to lack of space.

**Table 1.** Cardinality ( $\gamma$ ) of MDS and Time taken in seconds using Heuristic, Hedar, HGA and ACO-LS for UDG Instances

N	Range	Heuristic	Hedar		HGA		ACO-LS	
			$\gamma$	Time (s)	$\gamma$	Time (s)	$\gamma$	Time (s)
50	150	13.9	15.4	0.1	12.9	0.7	12.9	1.1
50	200	10.5	11.3	0.0	9.4	0.6	9.4	1.0
50	250	8	8.6	0.1	6.9	0.5	6.9	0.7
100	150	19.4	20.8	0.2	17	2.2	17	2.9
100	200	12.8	13.5	0.2	10.4	1.6	10.4	2.0
100	250	9.1	10.2	0.3	7.5	1.1	7.6	1.6
250	150	22.7	24.8	0.5	18.7	6.0	18.1	9.4
250	200	14.6	15.5	0.8	11.4	3.5	11	5.8
250	250	10.1	11.2	1.0	8	3.0	8	4.4
500	150	75.3	84.7	1.7	67.3	95.4	64.5	83.5
500	200	48.2	55.4	1.5	41.4	43.8	39.8	51.9
500	250	34.6	36.9	1.0	27.9	17.6	26.8	33.3
750	150	82.9	90.4	3.2	72.9	152.8	68.7	170.2
750	200	51.4	59	2.4	43.9	54.8	41.3	91.6
750	250	35.9	39.2	2.5	28.7	24.6	27.3	57.0
1000	150	85.9	94.2	4.4	74.8	215.1	70.3	264.3
1000	200	53	60	3.4	44.8	65.9	42.5	135
1000	250	36.7	39.8	4.0	29.8	35.5	28.2	83.9

## 4 Experimental Results

The experiments were done on two different types of graphs - unit disk graphs generated using the UDG topology generator [5] and Waxman Router Topologies [14] using

BRITE [7]. The data set consists of 50, 100, 250, 500, 750 and 1000 nodes. For UDG topologies, we used ranges of 150, 200 and 250 units to study the effect of different degrees of connectivity on the performance of the algorithms. Graphs with nodes 50, 100 and 250 are generated using an area of  $1000 \times 1000$  units whereas those with nodes 500, 750 and 1000 are generated using an area of  $2000 \times 2000$  units. In the case of Router Waxman topologies, we used random and heavy-tailed placement [6] which is considered more common for Internet topologies. We varied the degree of connectivity by using  $2 \times N$ ,  $4 \times N$  and  $8 \times N$  edges for graphs, where  $N$  is the number of nodes in the graph. In all cases, the results presented are averaged over 10 instances.

The hybrid genetic algorithm that we implemented (HGA) has an initial population of 100 members and we ran it for 10,000 generations. This would mean that a total of 10,000 solutions are created as we create one new member of the population in each generation. We use the following values for the various probabilities: when generating random population members both in initial population and a new member in later generations, we use a probability of 0.3 for adding a node into the dominating set. We use the value of  $p_c = 0.9$  for crossover,  $p_m = 0.02$  for mutation,  $p_{better} = 0.8$  to choose a better parent during binary tournament selection,  $p_h = 0.2$  for using the heuristic to repair a member. We use random removal of a redundant node with probability  $p_r = 0.6$ . All of these values were arrived at after extensive experimentation with different values. The hybrid ACO algorithm has 20 ants and is run for 1000 iterations which is a total of 20,000 solutions. We use an initial pheromone value of 10.0 and a minimum threshold on the pheromone value of 0.08. The pheromone persistence rate is  $\rho = 0.985$ . The algorithm from [4] has been used with 100 generations and 100 members in the population for a total of 10,000 solutions, the same as in our hybrid GA. The rest of the parameters are as specified in their paper.

It can readily be seen that the algorithm due to Hedar et. al. [4] is performing worse than even the greedy heuristic in all the graph instances studied. As stated earlier, if not for the changes we introduced, we were getting results that were even worse. This can be explained by the fact that there is no attempt to force each member of the population to be a DS. Minimization is also not done in the best possible way. The final *EliteInspiration* process seems flawed because in standard elitism, the best  $n_{core}$  members are always retained across generations. In their method, they are doing an intersection of the best  $n_{core}$  members which is not guaranteed to even be a DS.

We observe that the hybrid ACO performs better or on par with our own HGA for most instances in terms of cardinality of the solution. This shows that it is better to use ACO-LS for MDS than the HGA. We observe that the time taken by ACO-LS and HGA are similar up to 250 nodes for UDG instances. But, as the number of nodes increases, the ACO-LS algorithm takes more time than HGA, upto twice that of HGA. However, we note that the solutions generated by ACO-LS are also twice those generated by the GA. When it comes to large Router Waxman topologies, the time taken by ACO-LS is actually smaller than that of HGA. We also observe that for large Router Waxman graphs with more degree of connectivity, the HGA performs slightly better than the ACO-LS algorithm both in terms of cardinality as well as time.

Thus, we can conclude that the ACO-LS we have presented here is the best meta-heuristic approach found so far for the problem of MDS.

**Table 2.** Cardinality ( $\gamma$ ) of MDS and Time taken in seconds using Heuristic, Hedar, HGA and ACO-LS for Router Waxman Instances with random and heavy-tailed (ht) placement of nodes

N	Range	Placement	Heuristic	Hedar		HGA		ACO-LS	
				$\gamma$	Time (s)	$\gamma$	Time (s)	$\gamma$	Time (s)
50	100	ht	13.5	15.1	0.1	12.1	0.6	12.1	1.0
50	100	random	12.4	14.4	0.0	11.6	0.7	11.6	1.0
50	200	ht	7.7	10.9	0.1	7	0.5	7	0.6
50	200	random	7.3	10.3	0.0	6.8	0.5	6.8	0.8
50	400	ht	4.7	6.7	0.2	4.2	0.4	4.2	0.5
50	400	random	4.1	6.6	0.1	3.8	0.3	3.8	0.4
100	200	ht	26.7	31.8	0.1	23.6	2.4	23.7	3.4
100	200	random	25.8	32.1	0.2	23.4	2.7	23.4	3.3
100	400	ht	16.5	21.5	0.2	14.8	1.8	14.5	2.4
100	400	random	16.1	22.1	0.2	14.7	1.9	14.4	2.4
100	800	ht	9.6	15	0.1	8.7	1.1	8.7	1.6
100	800	random	9.2	15.5	0.2	8.7	1.2	8.4	1.6
250	500	ht	64.7	77.1	0.6	59.4	24.6	58.5	23.3
250	500	random	67.2	78.4	0.6	60.8	25.1	59.8	23.5
250	1000	ht	42.2	56.4	0.7	38.2	15.2	37.2	17.4
250	1000	random	41.8	57.9	0.6	38.5	16.2	37.1	17.4
250	1000	ht	26.1	39.6	0.5	23.7	7.4	23.1	11.1
250	1000	random	25.3	40.2	0.6	23.2	8.2	22.4	10.9
500	1000	ht	131.4	152.7	2.0	122.2	161.7	117.3	129.4
500	1000	random	131	157.7	2.0	121.7	162.2	117.9	127.2
500	2000	ht	83.9	113.5	2.3	78.4	94.5	75.8	97.9
500	2000	random	84.1	116.7	2.3	79.5	100.7	76.5	99.6
500	4000	ht	52.3	81.1	1.7	50.3	43.9	49	61
500	4000	random	50.1	83.8	1.7	48.7	49.1	47.2	60
750	1500	ht	196.6	230.8	4.2	183.7	520.5	176.9	375
750	1500	random	195.7	241.5	4.3	185.1	527.3	178.7	372.4
750	3000	ht	125.6	169.2	4.8	119.2	302.7	119.6	288.2
750	3000	random	127.3	176.4	4.6	120.8	319.2	119	290.4
750	6000	ht	78.2	120.5	4.1	76.6	134.9	77.7	171.2
750	6000	random	77.7	128.9	4.2	76.4	140.1	75.6	169.5
1000	2000	ht	268	317	7.5	251.3	1211.4	244.8	853.7
1000	2000	random	259.8	313.5	7.4	247.6	1212.3	239.1	824.7
1000	4000	ht	168.5	231.5	7.5	161.1	720.9	163.3	635.7
1000	4000	random	168	236.9	7.7	160.7	735.6	161.3	638.2
1000	8000	ht	104.1	166.3	8.5	103	300.7	106.5	359.7
1000	8000	random	104	171.4	8.2	102.5	311.1	106.7	367.8

## 5 Conclusions

We have proposed a steady-state hybrid genetic algorithm (HGA) and a hybrid ACO algorithm that combines ACO with local search (ACO-LS) for the problem of Minimum Dominating Set. We compared the results obtained by these two algorithms with

the greedy heuristic which is the optimal approximate solution and a hybrid genetic algorithm proposed by Hedar et. al. [4]. We find that the ACO-LS algorithm gives the best cardinality in almost all the cases. In most cases, its running time is comparable to that of the HGA and in some cases, it is actually better. It takes more time for large UDG instances compared to HGA. However, the ACO-LS algorithm generates twice as many solutions as HGA. This implies that the ACO-LS is actually much faster than the HGA. We found that an ACO which computes probability of a node for inclusion in the random walk of an ant based on both pheromone and a heuristic such as the degree of a node performs worse than the local search algorithm presented here. Thus, we conclude that the best solution is obtained using standard ACO enhanced with a minimization heuristic.

## References

1. Beasley, J.E., Chu, P.C.: A genetic algorithm for the set covering problem. *European Journal of Operational Research* 94(2), 392–404 (1996)
2. Chvatal, V.: A greedy heuristic for the set covering problem. *Mathematics of Operations Research* 4(3), 233–235 (1979)
3. Davis, L.: *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York (1991)
4. Hedar, A.R., Ismail, R.: Hybrid Genetic Algorithm for Minimum Dominating Set Problem. In: Taniar, D., Gervasi, O., Murgante, B., Pardede, E., Apduhan, B.O. (eds.) ICCSA 2010. LNCS, vol. 6019, pp. 457–467. Springer, Heidelberg (2010)
5. Mastrogiovanni, M.: *The clustering simulation framework: A simple manual* (2007), <http://www.michele-mastrogiovanni.net/software/download/README.pdf>
6. Medina, A., Lakhina, A., Matta, I., Byers, J.: Brite user manual, [http://www.cs.bu.edu/brite/user\\_manual/node42.html](http://www.cs.bu.edu/brite/user_manual/node42.html)
7. Medina, A., Lakhina, A., Matta, I., Byers, J.: Brite: An approach to universal topology generation. In: *Proceedings of the International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunications Systems, MASCOTS 2001* (2001)
8. Garey, M.R., Johnson, D.S.: *Computers and Tractability, A guide to the theory of NP-Completeness*. Freeman and Company, New York (1979)
9. Nieberg, T., Hurink, J.L.: A PTAS for the Minimum Dominating Set Problem in Unit Disk Graphs. In: Erlebach, T., Persinao, G. (eds.) WAOA 2005. LNCS, vol. 3879, pp. 296–306. Springer, Heidelberg (2006)
10. Shyu, S.J., Yin, P.Y., Lin, B.M.: An ant colony optimization algorithm for the minimum weight vertex cover problem. *Annals of Operation Research* 131(1-4), 283–304 (2004)
11. Singh, A., Baghel, A.S.: New metaheuristic approaches for the leaf-constrained minimum spanning tree problem. *Asia-Pacific Journal of Operational Research* 25(4), 575–589 (2008)
12. Solnon, C., Fenet, S.: A study of aco capabilities for solving the maximum clique problem. *Journal of Heuristics* 12, 155–180 (2006)
13. Wattenhoffer, R.: Distributed dominating set approximation, <http://www.disco.ethz.ch/lectures/ss04/distcomp/lecture/chapter12.pdf>
14. Waxman, B.: Routing of multipoint connections. *IEEE Journal of Selected Areas in Communication* 6(9), 1617–1622 (1988)

# Automatic Clustering Based on Invasive Weed Optimization Algorithm

Aritra Chowdhury<sup>1</sup>, Sandip Bose<sup>1</sup>, and Swagatam Das<sup>2</sup>

<sup>1</sup>Dept. of Electronics and Telecommunication Engg, Jadavpur University, Kolkata 700032, India

<sup>2</sup>Electronics and Computer Sciences Unit, Indian Statistical Institute, Kolkata, India  
arit0001@yahoo.co.in, sandip.bose2009@gmail.com,  
swagatam.das@ieee.org

**Abstract.** In this article, an evolutionary metaheuristic algorithm known as the Invasive Weed Optimization (IWO) is applied for automatically partitioning a dataset without any prior information about the number of naturally occurring groups in the data. The fitness function used in the genetic algorithm is a cluster validity index. Depending on the results of this index IWO returns the segmented dataset along with the appropriate number of divisions. The proficiency of this algorithm is compared to variable string length genetic algorithm with point symmetry based distance clustering(VGAPS-clustering), variable string length Genetic K-means algorithm(GCUK-clustering) and a weighted sum validity function based hybrid niching genetic algorithm(HNGA-clustering) and is denoted for the nine artificial datasets and four real life datasets.

**Keywords:** Invasive Weed Optimization, Clustering, Cluster validity index, Genetic Algorithm, Variable number of clusters.

## 1 Introduction

Clustering represents the partitioning an unlabeled dataset into groups of similar elements. Each group, known as a ‘cluster’, comprises of elements that are similar between themselves and dissimilar to objects of other groups.

It is necessary to define a measure of similarity or dissimilarity between objects of two groups in order to mathematically identify them as clusters in a dataset. This measure is entirely almost dependent on the information in the dataset which may be perceived as a guideline for assigning patterns to the domain of a particular cluster centroid.

Another pertinent consideration of clustering is the determination of the appropriate number of clusters from a given dataset, where the number of groups in the dataset is unknown *a priori*. There are many cluster indices which are proposed in literature which performs this function. The partition that results in the optimum value of the cluster validity function is selected as the true partitioning. The validity index used in this paper is Sym-k [1] which is based on point symmetry distance rather than Euclidean distance. Since the global optimum of the cluster validity functions

corresponding to the “best” solutions, stochastic clustering algorithms based on Genetic Algorithms (GAs) have been observed to optimize the validity functions to determine the cluster number and partitioning of the data simultaneously. In this paper, we use the Invasive Weed Optimization algorithm [2.3] to detect the number of clusters and evolve the correct partitioning of the dataset simultaneously. The basic concepts of seeding, growth and competition that are prevalent in a weed colony [2] are employed in this algorithm.

## 2 Basic Concepts

### 2.1 An Overview of Invasive Weed Optimization

Recently, in the literature, there has been considerable attention paid for using algorithms which are inspired from natural processes and/or events in order to solve optimization problems. The term ‘weed’ is reserved for those plants whose invasive habits of growth and reproduction serve as a threat to the cultivated plants. The concepts of seeding, growth and competition in a weed colony [2] is shown to be effective in converging to an optimal solution. The colonizing behavior of weeds may be simulated by outlining the following procedures:

- *Population Initialization*

A set of initial solutions are randomly dispersed over the  $d$ - dimensional search space. These initial solutions are analogous to seeds in the weed colony which represent the first generation of the population.

- *Reproduction*

A member plant in the population produces seeds based on its own fitness according to the lowest and the highest fitness of individuals in the population. The closer the fitness of the plant is to the highest fitness; correspondingly more number of seeds is produced. Such a type of population reproduction provides an opportunity of growth to infeasible individuals which may often carry more relevant information than the feasible individuals.

- *Spatial dispersal*

These seeds are randomly distributed over the  $d$  – dimensional search space by random numbers which are normally distributed. Even though, the mean of these numbers should be zero, the variance is variable. The standard deviation at every iteration is given by the following function:

$$sd_{iter} = \left( \frac{iter_{max} - iter}{iter_{max}} \right)^{pow} (sd_{max} - sd_{min}) + sd_{min}, \quad (1)$$



Where,  $sd_{iter}$  is the standard deviation at each iteration,  $sd_{max}$  and  $sd_{min}$  are the maximum and minimum standard deviations defined respectively.  $pow$  is a real number. Thus the probability of dropping a seed in a distant area decreases non-linearly with iterations. This is the selection mechanism of IWO [2].

- *Competitive exclusion*

Some kind of competition must exist for limiting the number of maximum and minimum plants in a colony. Initially, all plants will reproduce fast and all plants will be introduced in the colony, until the number of plants in the colony reaches a maximum value called  $pop_{max}$ . Only the fittest plants in the colony are selected as the population for the next generation.

## 2.2 Validity Index

The validity index used as a fitness function in the algorithm is the point symmetry based Sym-K index [4]. Point symmetry distance  $d_{ps}(\bar{x}, \bar{c})$  is evaluated for the point  $\bar{x}$  in the dataset relative to a cluster center  $\bar{c}$ .  $(2 \times \bar{c} - \bar{x})$  Which is denoted by  $\bar{x}^*$  is the point of symmetry of  $\bar{x}$  with respect to  $\bar{c}$ . Let  $k_{near}$  nearest neighbors of  $\bar{x}^*$  be at Euclidean distances of  $d_i$  s .  $i = 1, 2, \dots, k_{near}$ . We can define symmetric distance  $d_{sym}(\bar{x}, \bar{c})$  and  $d_{ps}(\bar{x}, \bar{c})$  as follows:

$$d_{sym}(\bar{x}, \bar{c}) = \frac{\sum_{i=1}^{k_{near}} d_i}{k_{near}} \tag{2}$$

$$d_{ps}(\bar{x}, \bar{c}) = d_{sym}(\bar{x}, \bar{c}) \times d_e(\bar{x}, \bar{c}), \tag{3}$$

where  $d_e(\bar{x}, \bar{c})$  is the Euclidean distance between points  $\bar{x}$  and  $\bar{c}$ .

Let us consider a segmentation of the dataset  $X = \{x_j, j = 1, 2, \dots, N\}$  where  $N$  is the number of individuals in the population.  $\bar{c}_i$  is the cluster center of the  $i$ th cluster given by :

$$\bar{c}_i = \frac{\sum_{j=1}^{n_i} x_j}{n_i}, \tag{4}$$

where  $n_i$  ( $i = 1, 2, \dots, K$ ) is the number of points in cluster  $i$  and  $\bar{x}_j^i$  denotes the  $j$ th point of the  $i$ th cluster. The cluster validity index  $Sym$  is defined as:

$$Sym(K) = \frac{1}{K} \times \frac{1}{\varepsilon_K} \times D_K, \quad (5)$$

where

$$\varepsilon_K = \sum_{i=1}^K E_i, \quad (6)$$

$$E_i = \sum_{j=1}^{n_i} d_{ps}^*(\bar{x}_j^i, \bar{c}_i), \quad (7)$$

and

$$D_K = \max_{i,j=1}^K \left\| \bar{c}_i - \bar{c}_j \right\|. \quad (8)$$

$D_K$  represents the maximum Euclidean distance between any two cluster centers among all the pairs of centers.  $d_{ps}^*$  is computed such that the first  $knear$  neighbors of  $\bar{x}_j^i = 2 \times \bar{c}_i - \bar{x}_j^i$  are among only those points which are in cluster  $i$ . In order to obtain the requisite number of clusters, this index should be maximized.

### 3 IWO - Automatic Clustering Algorithm

The proposed clustering algorithm is based on the methods of Invasive Weed optimization. The number of groups in the population is obtained automatically.

#### 3.1 Algorithm: IWO-Clustering

1. Randomly initialize the strings in a weed matrix consisting of  $n$  solutions. Each solution encodes for randomly selected  $K_i$  number of clusters.
2. The Sym-K validity indices of each solution are evaluated individually according to equations (2), (3), (4), (5), (6) and (7).
3. The standard deviation of the present iteration is computed according to equation (1).
4. The number of seeds of each plant (solution) in the present population is evaluated accordingly to the fitness values of each individual as formulated in step 2. These seeds are initially added to the already existing population.
5. If the number of individuals in the total population exceeds  $pop_{max}$  as defined by the user, only the fittest  $pop_{max}$  individuals are included while the rest of the undesirable ‘weeds’; are eliminated.
6. Steps 2 to 5 are repeated until a desired solution is achieved or the operation is terminated after a predefined number of generations.

### 3.2 Solution (Weed) Representation and Initialization of the Population

The weed matrix comprises of strings which are composed of real numbers encoding the centers of the proposed partitions.

Each weed in the population encodes a number of clusters  $K_i$ . Let  $K_{\max}$  and  $K_{\min}$  be the soft estimates upper and lower bound of the number of groups in the population. The number of clusters for each weed is evaluated as follows,

$$K_i = rand() \times (K_{\min} + (K_{\max} - K_{\min}))$$

Here,  $rand()$  is a function which returns an integer. For example, in a  $d$  - dimensional space, the weed  $i$  has a length of  $d \times K_i$ . A sample weed in a 3-dimensional space may be  $\langle 2.3 \ 4.4 \ 6 \ 7.82 \ 9.32 \ 4.2 \ 6.7 \ 6.8 \ 10.1 \ 11.2 \ 2.4 \ 5.9 \rangle$ . This weed accounts for 4 cluster centers (2.3, 4.4, 6), (7.82, 9.32, 4.2), (6.7, 6.8, 10.1) and (11.2, 2.4, 5.9).

## 4 Experimental Results

### 4.1 Datasets

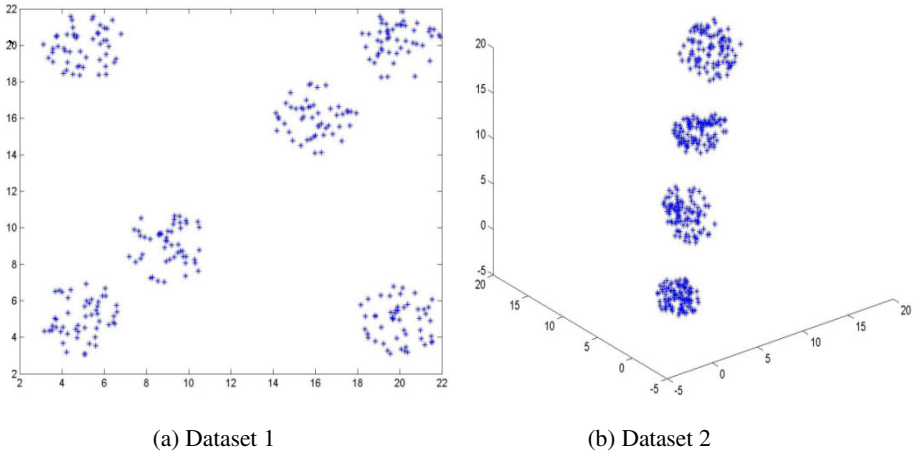
- **Artificial Datasets**

- (i) *Dataset1*: This dataset used in [4] consists of 300 points as represented Fig. 1(a). There are a total of 6 groups in the population. The clusters are all of the same size.
- (ii) *Dataset 2*: This is a 3 dimensional dataset. There are a total of 400 points in the data. There exist 4 spherical clusters as shown in Fig. 1(b).

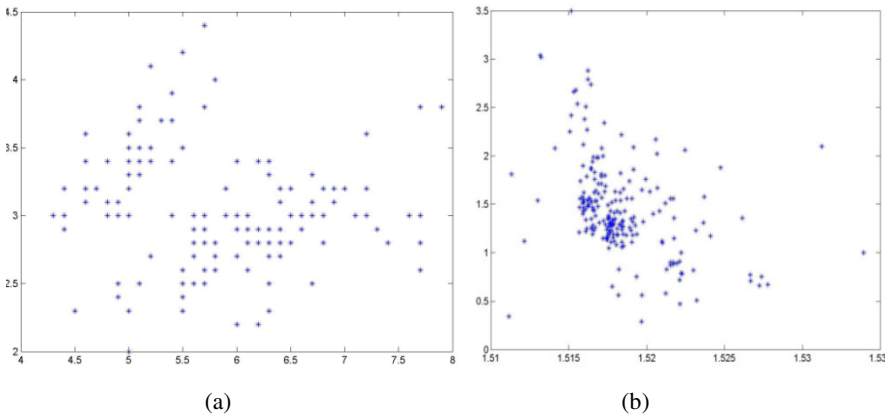
- **Real life Datasets** : Two real life data sets were obtained from [5] :

(i)*Iris*: It consists of 150 points equally distributed over 3 groups, viz. *Setosa*, *Versicolor* and *Virginica*. It is represented by 4 feature values. *Versicolor* and *Virginica* overlap while *Setosa* can be linearly separated. It is represented in 2 dimensions in Fig. 2(a).

(ii)*Glass*: This dataset has 9 features and consists of 214 instances. There are 6 groups in the dataset. It is shown in 2 dimensions in Fig. 2(b).



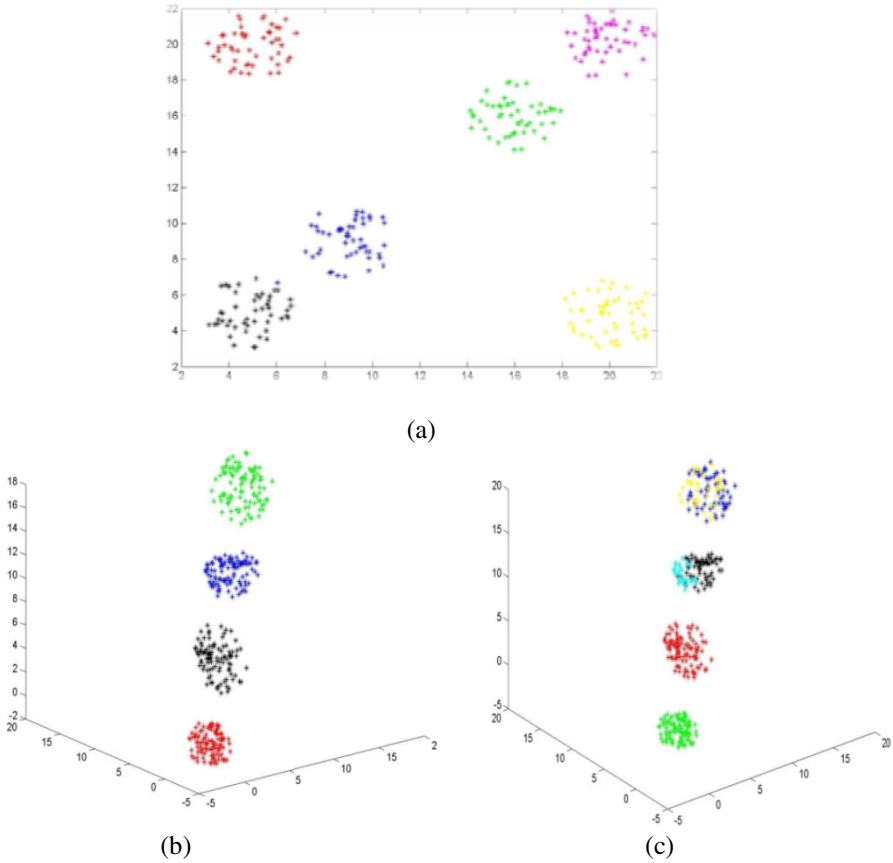
**Fig. 1.**



**Fig. 2.** (a) Iris dataset projected on the first and second dimension (b) Glass dataset projected on the first and fourth dimensions.

## 4.2 Results

This section compares the performance of the proposed clustering algorithm with a few automatic clustering algorithms like VGAPS [6], HNGA [7] and GCUK [4]. The contestant algorithms were run with the most appropriate of parameters obtained from their respective literature. The figures 3(a) ,3(b) are the clustered representations of the figures 2(a) and (b) obtained by IWO clustering algorithm. Figure 3(c) represents the clustered picturization after performing HNGA clustering on Dataset 2. Datasets 1 and 2 are artificial datasets and *Iris* and *Glass* datasets are real life datasets. Dataset 1 is a 2 dimensional dataset and Dataset 2 is a three dimensional dataset. These sets of data are used in order to represent all kinds of data adequately.



**Fig. 3.** (a) Clustered Dataset1 using IWO, VGAPS, GCUK and HNGA (b) Clustered Dataset 2 using IWO, VGAPS and GCUK resulting in 4 clusters (c) Clustered Dataset 2 using HNGA resulting in 6 clusters

### 4.3 Minkowski Score

Let T be the “true” solution and S be the solution obtained experimentally.

Minkowski score [8] is defined as:

$$MS(T, S) = \sqrt{\frac{n_{01} + n_{10}}{n_{11} + n_{10}}}, \quad (9)$$

where  $n_{01}$  represents the number of pairs of elements that are in the same cluster only in S and  $n_{10}$  represents the number of pairs of elements that are in the same cluster only in T.  $n_{11}$  denotes the number of pairs of elements that are in the same cluster both in S and in T.

**Table 1.** Minkowski Scores obtained for all the four algorithms for all the experimental datasets

Data Set	IWO Clustering	VGAPS Clustering	GCUK Clustering	HNGA Clustering
Dataset 1	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00
Dataset 2	0.00±0.00	0.00±0.00	0.00±0.00	0.2626±0.02
Iris	0.57±0.021	0.62±0.02	0.847726±0.01	0.854081±0.025
Glass	1.0519±0.013	1.106±0.01	1.324295±0.022	1.11794±0.023

## 5 Conclusion

From Table 1 we see that the Minkowski Scores obtained from equation (9) for IWO clustering are either less or equal to the scores for the other clustering algorithms. The effectiveness of this algorithm is evident from the Minkowski Scores of real life datasets *iris* and *glass*. The effectiveness of IWO clustering over VGAPS, GCUK and HNGA clustering is evident from the Minkowski scores of the real life datasets. The artificial datasets are seen to have perfect solutions as their Minkowski scores are zero. However these perfect solutions are obtained after only a minimum number of generations thus reducing the run time significantly.

## References

- [1] Bandyopadhyay, S., Saha, S.: A Point Symmetry Based Clustering Technique for Automatic Evaluation of Clusters. *IEEE Transactions on Knowledge and Data Engineering* 20(11) (November 2008)
- [2] Mehrabian, A.R., Lucas, C.: A Novel Optimization Algorithm Inspired from Weed Colonization. In: *Ecological Informatics*. Elsevier (2006)
- [3] Sepehri Rad, H., Lucas, C.: A Recommender System Based On Invasive Weed Optimization Algorithm. In: *IEEE Congress on Evolutionary Computation, CEC 2007*, pp. 4297–4304 (2007)
- [4] Bandyopadhyay, S., Maulik, U.: Genetic Clustering for Automatic Evolution of Clusters and Application to Image Classification. *Pattern Recognition* (2), 1197–1208 (2002)
- [5] D.N.A. Asuncion: *UCI Machine Learning Repository* (2007)
- [6] Holland, J.H.: *Adaptation in Natural and Artificial System*. The University of Michigan Press, AnnArbor (1975)
- [7] Sheng, W., Swift, S., Zhang, L., Liu, X.: A Weighted Sum Validity Function for Clustering with a Hybrid Niching Genetic Algorithm. *IEEE Transactions on Systems, Man and Cybernetics – Part B: Cybernetics* 35(6) (December 2005)
- [8] Ben-Hur, A., Guyon, I.: *Detecting Stable Clusters Using Principal Component Analysis in Methods of Molecular Biology*. Humana Press (2003)

# Classification of Anemia Using Data Mining Techniques

Shilpa A. Sanap<sup>1</sup>, Meghana Nagori<sup>2</sup>, and Vivek Kshirsagar<sup>2</sup>

<sup>1</sup> Department of Computer Science and Engineering, Marathwada Institute of Technology,  
Dr. B.A.M. University, Aurangabad Maharashtra, India  
shilpa.sanap@gmail.com

<sup>2</sup> Department of Computer Science and Engineering, Govt. Engineering College,  
Dr. B.A.M. University, Aurangabad Maharashtra, India  
{kshirsagarmeghana, vkshirsagar}@gmail.com

**Abstract.** The extraction of hidden predictive information from large databases is possible with data mining. Anemia is the most common disorder of the blood. Anemia can be classified in a variety of ways, based on the morphology of RBCs, etiology, etc. In this paper we present an analysis of the prediction and classification of anemia in patients using data mining techniques. The dataset constructed from complete blood count test data from various hospitals. We have worked out with classification method C4.5 decision tree algorithm and Support vector machine which are implemented as J48 and SMO(sequential minimal optimization) in Weka. Several experiments are conducted using these algorithms. The decision tree for classification of anemia is generated which gives best possible classification of anemia based on CBC reports along with severity of anemia. We have observed that C4.5 algorithm has best performance with highest accuracy.

## 1 Introduction

This Data mining is a relatively new field of research whose major objective is to acquire knowledge from large amounts of data. In medical and health care areas, due to regulations and due to the availability of computers, a large amount of data is becoming available. There are a number of data mining algorithms and tools available we consider data mining tool Weka[8] to evaluate on the data. With classification techniques learning algorithms take a set of classified examples (training set) and use it for training the algorithms. With the trained algorithms, classification of the test data takes place based on the patterns and rules are extracted from the training set. We have used this strategy for prediction and classification of anemia. Anemia is defined as a reduction in the number of circulating red blood cells, the hemoglobin concentration, or the volume of packed red cells (hematocrit) in the blood.. A doctor can determine if you are anemic by performing a routine blood test called a complete blood count (CBC) test, which provides levels for both hemoglobin and hematocrit (the percentage of red blood cells in a blood sample). Anemia is best defined & monitored by measurement of hemoglobin concentration.[2]. The classification of anemia can be done by using various classification schemes like morphological, cytometric, erythrokinetic, pathogenic etc. Severity of anemia is categorized as mild,

moderate or severe depending how far a patient's hemoglobin level resides below normal range. In this paper we have used data mining classification methods for prediction and best possible classification anemia in different types along with severity levels. For classification of anemia cytometric and morphological schemes are used along with RDW-red cell distribution width.

## 2 Theoretical View

### 2.1 Decision Tree

A decision tree is a flow-chart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and leaf nodes represent classes or class distributions.[6] A decision tree is a tree in which each branch node represents a choice between a number of alternatives, and each leaf node represents a decision.

### 2.2 C4.5 Decision Tree Algorithm

C4.5[9] is an algorithm used to generate a decision tree developed by Ross Quinlan. C4.5 is an extension of Quinlan's earlier ID3 algorithm. The decision trees generated by C4.5 can be used for classification. C4.5 builds decision trees from a set of training data in the same way as ID3, using the concept of information entropy. The training data is a set  $S = s_1, s_2, \dots$  of already classified samples. Each sample  $s_i = x_1, x_2, \dots$  is a vector where  $x_1, x_2, \dots$  represent attributes or features of the sample. The training data is augmented with a vector  $C = c_1, c_2, \dots$  where  $c_1, c_2, \dots$  represent the class that each sample belongs to. C4.5 uses the fact that each attribute of the data can be used to make a decision that splits the data into smaller subsets. C4.5 examines the normalized information gain (difference in entropy) that results from choosing an attribute for splitting the data. The attribute with the highest normalized information gain is the one used to make the decision. The algorithm then recurs on the smaller sublists. Decision trees represent a supervised approach to classification.

### 2.3 Support Vector Machine

The support vector machine (SVM) is a recently developed technique for multidimensional function approximation. The objective of support vector machines is to determine a classifier or regression function which minimizes the empirical risk (that is, the training set error) and the confidence interval (which corresponds to the generalization or test set error)[17] SMO implements the sequential minimal optimization algorithm for training a support vector classifier using polynomial or Gaussian kernels. Support vector machines (SVMs) are a set of related supervised learning methods used for classification and regression. In simple words, given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that predicts whether a new example falls into one category or the other. Intuitively, an SVM model is a representation of the examples as points in



space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

## 2.4 Weka

The WEKA software was developed in the University of New Zealand and the name stands for Waikato Environment for Knowledge Analysis. A number of data mining methods are implemented in the WEKA software. Some of them are based on decision trees like the J48 decision tree, some are rule-based like ZeroR and decision tables and some of them are based on probability and regression, like the Naïve bayes algorithm. Weka provides a uniform interface to many different learning algorithms, along with methods for pre- and postprocessing and for evaluating the result of learning schemes on any given dataset. All algorithms take their input in the form of a single relational table in the ARFF format which can be read from a file or generated by a database query. Weka's native storage method is ARFF format. So conversion has been performed to make the examination data available for analysis through Weka.

## 2.5 Classification of Anemia

### 2.5.1 Anemia Definition

Anemia is a medical condition in which the red blood cell count or hemoglobin is less than normal. In the laboratory, anemia is identified when a patient's hemoglobin (HGB)/hematocrit (HCT) values fall below the lower end of a normal range of values for age- and sex-matched subjects.

### 2.5.2 Diagnosis of Anemia with CBC

Anemia is usually detected or at least confirmed by a complete blood cell (CBC) count[5]. CBC test may be ordered by a physician as a part of routine general check-up. Traditionally, CBC analysis was performed by a physician or a laboratory technician by viewing a glass slide prepared from a blood sample under a microscope, performed on an automated hematology analyzer using well mixed whole blood that is added to a chemical called EDTA (ethylenediaminetetracetic acid)[16] to prevent clotting.

### 2.5.3 Classification of Anemia Using CBC Reports

There are several types of anemia. From the CBC report, one can classify anemia as microcytic, normocytic or macrocytic if the MCV(Mean corpuscular volume) is low, normal or high, respectively. Anemia can be classified as normochromic and hypochromic if MCHC (Mean corpuscular hemoglobin concentration) is normal or high respectively. Anemia can be classified by cytometric schemes (i.e., those that depend on cell size and hemoglobin-content parameters, such as MCV and MCHC),[4] as 1. Normochromic, normocytic anemia which include anemia of chronic disease(90%), hemolytic anemia, anemia of acute hemorrhage, aplastic anemia 2. Hypochromic, microcytic anemia includes iron deficiency anemia, thalassemia 3. Normochromic, macrocytic anemia includes vitamin B12 deficiency folate deficiency

4. Microcytic, normochromic anemia [12] include Anemia of renal disease RDW is a very useful measure in the assessment of anemia. Combined with red cell indices, it can narrow down the diagnostic possibilities. For example, a patient with microcytic anemia and high RDW is very likely to have iron deficiency. If the RDW is normal thalassemia become much more likely [7].

### 3 Methodology

Our aim is to investigate C4.5 decision tree algorithm and support vector machine (SMO in weka) and to find which method gives most suitable technique for prediction and best possible classification of anemia using CBC data and generate decision tree. We have collected data from complete blood count tests which are performed by collecting blood samples from 191 healthy individuals and 323 patients having disorders of anemia. The data contains 514 instances.

#### 3.1 Data Preprocessing and Data Cleaning

In this step we try to eliminate noise that is present in the data. Noise can be defined as some form of error within the data. For example missing values duplicates. Preprocessing the input data set for a knowledge discovery goal using a data mining approach usually consumes the biggest portion of the effort devoted in the entire work. The data is collected from complete blood count test reports and dataset CBC\_ANEMIA is constructed.

#### 3.2 Attribute Selection

Attribute selection include selecting relevant attributes and removal of redundant and/or irrelevant attributes. The CBC data contains 22 attributes irrelevant attributes are removed . The relevant attributes are shown in table 1.

**Table 1.** Numeric attributes of CBC\_Anemia dataset with reference ranges

Numeric attribute	Mean	StdDev	Ref. Range
AGE	30	18	1-100
WBC-White blood cell	9.35	4.57	4.0-11.0
HGB-Hemoglobin	10.87	2.71	11.0-16.0 g/dL
RBC-Red blood cell	4.43	0.89	3.50-6.20
HCT-Hematocrit	34.66	7.95	37.0-50.0
MCV- Mean corpuscular volume	78.79	11.9	82.0-95.0 fL
MCH Mean corpuscular hemoglobin	24.52	4.47	27.0-31.0 pg
MCHC- Mean corpuscular hemoglobin concentration	31.06	1.83	32.0-36.0%
RDW-Red cell distribution width	15.91	2.1	11.5-16.5%
PLT-Platelet	243.3	97.5	150-450

The dataset with selected attribute is converted into .arff file which can be recognized by Weka data mining tool[15]. The arff file contains numeric attributes which are mainly hematological parameters and some nominal attributes. For example parameter MCV is used to classify anemia as microcytic if its value is below reference range ( $MCV < 80$ ), as normocytic if MCV is in between 80-100 and as macrocytic if  $MCV > 100$ . The classification of anemia is done with consideration of severity of anemia. Severity of anemia is decided as severe, moderate and mild depending on value of hemoglobin.

### 3.3 Feature Extraction and Classification

The data collected is being processed in this step. The data is classified using C4.5 decision tree algorithm. Anemia can be classified based on MCV as Microcytic,

**Table 2.** Rule base for classification of anemia

Rule	Decision
A if(Detect_anemia=Severe)and(MCV=Micro)and(MCHC=Hypo) and (RDW-CV>16.5)then	G1 IDA
B else if(Detect_anemia=Severe)and(MCV=Normo)and (RDW-CV>16.5)then	G1 Cd/SS
C else if(Detect_anemia=Severe)and(MCV=Micro)and(MCHC=Normo) then	G1 ARD
D else if(Detect_anemia=Severe)and(MCV=Normo)and (RDW-CV<=16.5)then	G1 ACD
E else if(Detect_anemia=Severe)and(MCV=Macro)and (RDW-CV<=16.5)then	G1 APA
F else if(Detect_anemia=Severe)and(MCV=Micro)and(MCHC=Hypo)and (RDW-CV<=16.5)then	G1 THAL
G else if(Detect_anemia=Moderate)and(MCV=Micro)and(MCHC=Hypo)and (RDW-CV>16.5)then	G2 IDA
H else if(Detect_anemia=Moderate)and(MCV=Micro)and(MCHC=Hypo)and (RDW-CV<=16.5)then	G2 THAL
I Elseif(Detect_anemia=Moderate)and(MCV=Macro)and (RDW-CV>16.5)then	G2 BFD
J Elseif(Detect_anemia=Moderate)and(MCV=Normo)and (RDW-CV<=16.5)then	G2 ACD
K Elseif(Detect_anemia=Moderate)and(MCV=Normo)and (RDW-CV>16.5)then	G2 Cd/SS
L Elseif(Detect_anemia=Moderate)and(MCV=Macro)and (RDW-CV<=16.5)then	G2APA
M Elseif(Detect_anemia=Moderate)and(MCV=Micro)and(MCHC=Normo)then	G2 ARD
N Elseif(Detect_anemia=Mild)and(MCV=Micro)and(MCHC=Normo)then	G3 ARD
O Elseif(Detect_anemia=Mild)and(MCV=Normo)and (RDW-CV>16.5)then	G3 Cd/SS
P Elseif(Detect_anemia=Mild)and(MCV=Normo)and (RDW-CV<=16.5)then	G3 ACD
Q Elseif(Detect_anemia=Mild)and(MCV=Micro)and(MCHC=Hypo)and (RDW-CV<=16.5)then	G3 THAL
R Elseif(Detect_anemia=Mild)and(MCV=Micro)and(MCHC=Hypo)and (RDW-CV>16.5)then	G3 IDA
S Elseif(Detect_anemia=Mild)and(C-MCV=Macro)and (RDW-CV>16.5)then	G3 BFD
T Elseif(Detect_anemia=Normal) and (GENDER=F)then	N. female
U Elseif(Detect_anemia=Mild)and(C-MCV=Macro)and (RDW-CV<=16.5)then	G3 APA
V else if(Detect_anemia=Normal) and (GENDER=M)then	N. male

Normocytic & Macrocytic and as Normochromic & Hypochromic based on values of MCHC. The patients having microcytosis and hypochromic are divided into subgroups iron deficiency anemia(IDA) and thalassemia(THAL)[14]. The normocytic patient can further be categorized as anemia of chronic disease in 90% cases and as combined deficiency anemia or sickle cell disease (CDorSS) depending on red cell distribution width. Further patient having macrocytosis can be classified as vitamin B12 or folate acid deficiency anemia(BFD) in majority of cases or as aplastic anemia(APA) if red cell distribution width is normal[11]. Decision tree is constructed from the data using C4.5 decision tree algorithm Table 2 contains the decision tree index and rule corresponding to the decision made in decision tree.

### 3.4 Evaluation

Evaluation of data is done in by using all 514 instances of data using C4.5 decision tree algorithm(J48 in weka) and support vector machine(SMO in weka) with test options Mode1- use training set data Mode2-cross validation 4- fold and Mode3-cross validation 10-fold. The results of evaluation on CBC\_Anemia dataset with J48 algorithm and SMO through Weka are shown in Table 3 .This table depicts accuracy along with correctly and F-measure using various test options[1]. The evaluation is done using all 514 instances.

**Table 3.** Evaluation on CBC\_Anemia data using C4.5 algorithm and SVM

Testing Criterion	C4.5 Decision Tree(J48 ) Algorithm		Supportvector Machine(SMO) Algorithm	
	Correctly classified instances	F-measure	Correctly classified instances	F-measure
Training set data	99.42%	0.993	88.13%	0.837
Cross validation 2-fold	96.89%	0.964	86.77%	0.816
Cross validation 4-fold	97.08%	0.968	87.35%	0.826
Cross validation 10-fold	97.67%	0.974	87.35%	0.824

Mode1 –Training set data uses all 514 records as training data which trains the model for classification and gives accuracy of 99.42% with J48 and of 88.13% with SMO. Further F-measure for J48 is 0.993 that for SMO is 0.837.

In cross-validation, you decide on a fixed number of folds, or partitions of the data Mode2-cross validation 4-fold. In which data is split into four approximately equal partitions and each in turn is used for testing and the remainder is used for training. That is, use three-fourth for training and one-fourth for testing and repeat the procedure four times so that, in the end, every instance has been used exactly once for testing.Mode3-cross validation 10-fold. The standard way of predicting the error rate of a learning technique given a single, fixed sample of data is to use stratified 10-fold cross-validation. The data is divided randomly into 10 parts in which the class is represented in approximately the same proportions as in the full dataset. Each part is

held out in turn and the learning scheme trained on the remaining nine-tenths; then its error rate is calculated on the holdout set. Thus the learning procedure is executed a total of 10 times on different training sets. Finally, the 10 error estimates are averaged to yield an overall error estimate. Whereas for Mode 2 used 1/10 of total records for testing the model created and kept on repeating the process for the second portion of 1/10 and so forth until 10 times. The comparison of the accuracy and F-measure on the dataset between J48 and SMO are illustrated in Figures 1.

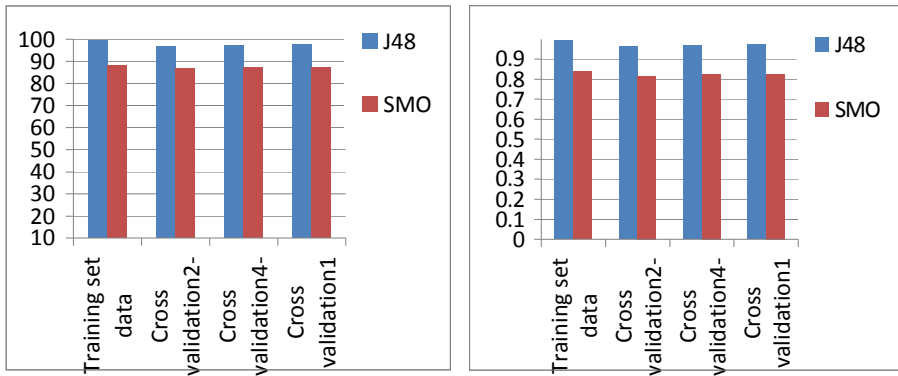


Fig. 1. Comparing accuracy F-measure J48 & SMO respectively

## 4 Conclusion

Following conclusions can be drawn

1. C4.5 decision tree algorithm classify anemia perfectly with accuracy of 99.42 %,there are only 3 instances classified incorrectly out of 514 instances and support vector machine with accuracy of 88.13% for the same.
2. C4.5 algorithm is best classification technique with highest accuracy and F-measure to generate decision tree for classification of anemia based on CBC test data.
3. Decision Trees, as a data mining technique, is very useful in the process of knowledge discovery in the medical field. In addition, using this technique is very convenient since the Decision Tree is simple to understand, works with mixed data types, models non-linear functions, handles classification, and most of the readily available tools use it.
4. Using the same data sets with different mining techniques and comparing results of each technique in order to construct a full view of the resulted patterns and levels of accuracy of each technique may be very useful for this application.
5. Data mining classification techniques can provide assistance in making the diagnosis or classification of anemia based on complete blood count, but it still cannot replace the physician's intuition and interpretive skills. The resulted classification reacted as a guidance to monitor future undertakings in order to control them from being far away from the appropriate classification of anemia.

**Acknowledgment.** Our special thanks to Mahatma Gandhi Mission Medical Center & Research Institute & Dr. Hedgewar Rugnalaya Central Laboratory for the clinical data sets and Dr. Prabhakar T. Sanap , Dr. Bhale and Dr. R.S. Patwadkar for their invaluable expertise in medication and procedures in support of this work.

## References

- [1] Razali, A.M., Ali, S.: Generating Treatment Plan in Medicine: A Data Mining Approach. American Journal of Applied Sciences 6(2), 345–351 (2009), ISSN 1546- 9239 © 2009 Science Publications
- [2] Schmaier, A.H., Petruzzelli, L.M.: Hematology for the medical student. Lippincott Williams and Wilkins 25
- [3] Bernadette, F.R., Doig, K.: Hematology: Clinical features & applications, 3rd edn., pp. 227–230
- [4] Ed Uthman’s homepage; Anemia Pathophysiologic Consequences, Classification, and Clinical Investigation (2009),  
<http://web2.airmail.net/uthman/anemia/anemia.html>
- [5] Fischbach, F.T.: A manual of laboratory & diagnostic tests, 6th edn. (2008)
- [6] Han, J., Kamber, M.: Data Mining Concepts and Techniques, 2nd edn. (2001)
- [7] Hematology complete blood count,  
[http://www.meddean.luc.edu/lumen/MedEd/MEDICINE/medclerk/2004\\_05/level11/CBCAnemia/cbc\\_f.htm](http://www.meddean.luc.edu/lumen/MedEd/MEDICINE/medclerk/2004_05/level11/CBCAnemia/cbc_f.htm)
- [8] Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Fransisco (2005)
- [9] Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo (2003)
- [10] Cadez, L.V., MacLaren, C.E., Smyth, P., McLachlan, G.J.: Hierarchical model for screening Iron Deficiency Anemia. Technical report no 99-14, Department of Information and Computer Science, University of California, Irvine
- [11] Wood, M.E., Philips, G.K.: Hematology/oncology secrets, 3rd edn., pp. 20–21
- [12] Medical Technology; RBC indices and anemia classification,  
<http://www.irvingcrowley.com/cls/anemia.htm>
- [13] Practical Utilization of the Complete Blood Count. Joseph M. Flynn, D.O.,MPH, FACP. Division Hematology-Oncology. THE Ohio State University, Columbus, OH (April 2008),  
[http://sciocountrymedicalsociety.org/documents/CBC\\_Flynn.PPT](http://sciocountrymedicalsociety.org/documents/CBC_Flynn.PPT)
- [14] Ravel, R.: Clinical laboratory medicine: Clinical application of laboratory data, 6th edn., pp. 13–14 (1993)
- [15] Weka: Data Mining Software in Java,  
<http://www.cs.waikato.ac.nz/ml/weka/>
- [16] Beck, W.S.: Hematology, 5th edn., pp. 604–613
- [17] Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning 20, 273–297 (1995)



# Taboo Evolutionary Programming Approach to Optimal Transfer from Earth to Mars

M. Mutyalarao, A. Sabarinath, and M. Xavier James Raj

Applied Mathematics Division, Vikram Sarabhai Space Centre,  
Trivandrum, India 695022

{m\_mutyalarao,a\_sabarinath,m\_xavierjamesraj}@vssc.gov.in

**Abstract.** Taboo evolutionary programming (TEP) is a novel evolutionary programming technique found extensive usage in the present decade. The algorithm can be implemented in many complex problems in science and technology to find the optimum solutions with constraints. In this paper, we studied a two-point boundary value problem such as Lambert conic determination to find out the optimum impulsive requirements for Earth to Mars transfer from a Geo stationary Transfer Orbit (GTO). The TEP results were compared with Genetic Algorithm (GA) and found that the TEP gives better results.

**Keywords:** Evolutionary Programming, Genetic algorithm, optimum, flyby, orbiter, ballistic trajectory.

## 1 Introduction

As operational costs to be reduced, space system engineers are facing the challenging task of minimizing the payload-launch mass ratio, while achieving the primary mission goals. To meet these requirements during the last two decades, global optimization approaches have been extensively used towards the solution of complex interplanetary trajectory transfers. Many authors proposed different methods of optimization techniques and tested on variety of cases [1-2]. In the recent years, stochastic optimization techniques have seen a remarkable development. There has been an enormous increase in the successful use of these techniques for many practical problems in the fields of science, technology, economics, logistics, and travel scheduling etc., which involve global optimization. In many cases it is shown, the ability to efficiently explore the solutions and a number of good initial guesses that have been further refined through the use of more accurate local optimization technique. The difficulty in global optimization increases with the dimension of the problem (defined as the number of variables involved) and the presence of multiple local minima. Further any effective global optimization technique must be able to avoid entrapment in local minima and to continue the search to give the optimal solution independent of the initial conditions.

In this paper an initial analysis of the effectiveness and usefulness of Taboo evolutionary programming (TEP) to a test case of direct ballistic transfer from Earth



to Mars is presented. The methodology is integrated with TEP to find the minimum energy opportunity (or minimum impulsive velocity requirement) transfer trajectories from Earth to Mars. The methodology is also integrated with a simple genetic algorithm (GA) and compared with TEP result.

## 2 Taboo Evolutionary Programming

### 2.1 Taboo Search (TS)

In 1995, Taboo (or 'Tabu' being a different spelling of the same word) Search (TS), originally developed by Glover [3-4] and extended to continuous valued functions, is a stochastic optimization method attracted much attention. A study with a number of benchmark test examples covering constrained and unconstrained functions was carried out by Rajesh et al [5] along with a rigorous comparison of the performance of the TS with other methods. The convergence of TS for continuous function optimization is studied by Mingjun Ji & Jacek Klinowski [6-7]. The results clearly revealed that TS technique can be viable alternative to other methods such as GA, Simulated annealing based on stochastic differential equations, pure random search, etc.

TS is a mathematical optimization method, belonging to the class of local search techniques. TS enhance the performance of a local search method by using memory structures: once a potential solution has been determined, it is marked as "Taboo" so that the algorithm does not visit that possibility repeatedly. TS uses a local or neighborhood search procedure to iteratively move from a solution  $x_i$  to another solution  $x_{i+1}$  in the neighborhood of  $x_i$ , until stopping criterion has been satisfied. Here the admitted solution in the neighborhood of  $x_i$ ,  $N^*(x_i)$ , are determined through the use of a memory structures, called *Taboo list*. The *Taboo list* is a short-term memory which contains the solutions that have been visited in the recent past. The search then progresses by iteratively (using Taboo list) moving from a solution  $x_i$  to a solution  $x_{i+1}$  in  $N^*(x_i)$ . A condition that guides the search to get out from the local optimum is called the *Taboo condition*.

### 2.2 Evolutionary Programming (EP)

An important branch of evolutionary algorithms (EAs) is evolutionary programming (EP) which attracted much attention for the determination of the global optimum of a specified function. EP is non-gradient algorithm and it uses primarily search based methodology to compute the optimum of a function. Other branches of EAs include GA and evolution strategies.

In 2006, a new method of global optimization technique, Taboo evolutionary programming motivated by TS and EP was first introduced by Mingjun Ji & Jacek Klinowski [6-7]. TEP essentially combines the features of an EP, called single-point mutation published by Ji. et al [8], with TS. The results were found to be in good agreement with that of analytical results.

### 2.3 TEP Algorithm

The TEP algorithm used in the present study is closely follows that of [6]. The objective is to compute the minimum of  $f(x)$  such that  $x \in \Omega$ , where  $\Omega = \{x \in R^n: a \leq x(j) \leq b, a, b \in R, j = 1, 2, \dots, n\}$ ,  $f$  is a real-valued continuous function on  $\Omega$ .

- 1) Generate the initial population of  $\mu$  individuals based on a uniform distribution, and set  $k = 1$ . Each individual is taken as a vector  $x_i, \forall i \in \{1, 2, \dots, \mu\}$
- 2) evaluate the fitness score for each individual,  $x_i, \forall i \in \{1, 2, \dots, \mu\}$ , of the population on the objective function,  $f(x_i)$ .
- 3) Each parent  $x_i, \forall i \in \{1, 2, \dots, \mu\}$ , create a single offspring  $x'_i$  by
 
$$x'_i(j_i) = x_i(j_i) + \eta N_i(0, 1), \quad \eta = \eta \exp(-\alpha),$$
 where  $j_i$  is randomly chosen from the set  $\{1, 2, \dots, n\}$  and the other components of  $x'_i$  are equal to the corresponding  $x_i$ 's.  $N(0, 1)$  denotes a normally distributed one-dimensional random number with a mean of zero and a standard deviation of one. Here, the parameter  $\alpha = 1.01$ . Initial value of  $\eta$  is  $\frac{b-a}{2}$  and whenever  $\eta < 10^{-4}$  then  $\eta$  set to its initial value.
- 4) Calculate the fitness of each offspring  $x'_i, \forall i \in \{1, 2, \dots, \mu\}$ .
- 5) Perform the search using the following improved paths:
  - 5.1 Choose an improved path as follows. For each  $i \in \{1, 2, \dots, \mu\}$ , if  $f(x'_i) \leq f(x_i)$ , then  $y_i = x'_i, d_i = x'_i - x_i$  is an improved path. Put a pair of vectors  $(y_i, d_i)$  into the set  $A$ .
  - 5.2 Choose  $r$  best fitness individuals from the set  $A$  as the parents with improved paths. Note that  $(y_m, d_m), m = 1, 2, \dots, r$ , where  $y_m$  is an objective variable and  $d_m$  is the corresponding improved path. Set  $A = \emptyset$ .
  - 5.3 Calculate fitness : for each  $m = 1, 2, \dots, r, y'_m = y_m + \rho d_m, \rho = \rho e^{-\alpha}$ . Initial value of  $\rho$  is 1 and whenever  $\rho < 10^{-6}$ , then  $\rho = 1$ .
  - 5.4 For each  $m = 1, 2, \dots, r$ , if  $f(y'_m) \leq f(y_m)$ , then set  $(y'_m, d_m)$  as a parent of the next generation with improved search paths, and put into the set  $A$ .
  - 5.5 Record the number  $\tau$ , of members in set  $A$ .
- 6) Choose the parents for the next generation.
  - 6.1 Perform a comparison over the union of parents  $x_i$  and offspring  $x'_i, \forall i \in \{1, 2, \dots, \mu\}$ . For each individual,  $q$  opponents are chosen uniformly at random from all the parents and offspring. For each comparison, if the individuals fitness is equal to or greater than the opponent's, it scores a 'win'. Select the  $\mu - \tau$  individuals out of  $x_i$  and  $x'_i, \forall i \in \{1, 2, \dots, \mu\}$ , which have the most wins to be put into the set  $B$ .
  - 6.2 Make the individuals  $x, x'$  and  $y$  from sets  $B$  and  $A$  the parents of the next generation. Set  $B = \emptyset$ .

- 7) Check the Taboo status
  - 7.1 Record the current optimal fitness,  $f_k^*$ , and the current optimal solution,  $x_k^*$ .
  - 7.2 When  $k > L$ , compare the optimal fitness of the current generation with the optimal fitness of the previous  $L$  generations. Thus, if  $|f_k^* - f_{k-L}^*| \leq \sigma_1$ , ( $\sigma_1$  is arbitrarily small value) then  $f^* = f_k^*$ ,  $x^* = x_k^*$ . Put the pair of vectors  $(f^*, x^*)$  into the taboo table  $\psi$ . The length of taboo table  $\psi$  is  $l$ .
  - 7.3 For any  $(f^*, x^*)$  in  $\psi$ , if the current optimal fitness  $f_k^*$  and the optimal solution  $x_k^*$  satisfies the taboo conditions,  $|f_k^* - f^*| \leq \sigma_1$  and  $\|x_k^* - x^*\| \leq \sigma_2$ , then generate the initial population of  $\mu$  individuals and set new individuals as the  $k^{th}$  generation.
- 8) Terminate if the halting criterion is satisfied. Otherwise, set  $k = k + 1$  and go to Step-3. In the present study the halting criterion used is the number of generations.

### 3 Problem Formulation

Minhazul Islam. et al [9] studied space trajectory optimization problem using an adaptive differential evolution algorithm with novel mutation and crossover strategies. In the present paper, optimization problem related to the interplanetary transfer trajectories is studied. In particular the problem to compute the minimum energy opportunity for a direct ballistic transfer from Earth to Mars is considered. The energy is calculated in terms of sum of all instantaneous velocity changes during the mission. Therefore the problem is to find the values of design variables which results the minimum total velocity ( $\Delta V$ ). The constraints considered in this paper include co-orbital plane maneuvers and the search space is characterized by three design variables. viz., departures date (DD), transfer duration ( $\Delta t$ ), and the orbital inclination ( $i$ ).

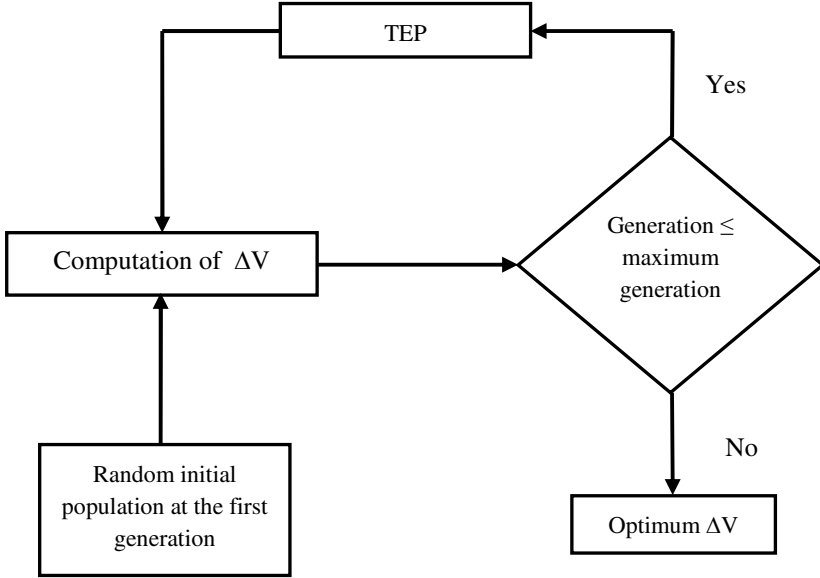
### 4 Methodology

The methodology used in the present study is presented in Fig. 1. The segment computation of  $\Delta V$  in Fig. 1 is explained below.

#### 4.1 Computation of $\Delta V$

Given the departure date (DD), time of flight ( $\Delta t$ ) and inclination ( $i$ ) of the orbit, the steps involved in the calculation of impulsive  $\Delta V$  is given as follows

- Compute the state vector  $(\vec{r}_1, \vec{v}_1)$  of the departure planet Earth at the departure date (DD).
- Compute the state vector  $(\vec{r}_2, \vec{v}_2)$  of the arrival planet Mars at the arrival date (i.e., DD+ $\Delta t$ ). In this paper the planetary ephemerides were modeled using Meeus algorithm given in Vallado [10].



**Fig. 1.** Schematic diagram of methodology

- Use Lambert problem solution technique (universal variable method Battin [11]) for the departure to target phase and determine initial ( $\vec{v}_1$ ) and final ( $\vec{v}_f$ ) vectors of the transfer hyperbola.
- Compute the asymptotic relative velocity vector at departure ( $\vec{v}_{\infty D}$ ) and at the arrival ( $\vec{v}_{\infty A}$ ) by using the following formulae:

$$\vec{v}_{\infty D} = \vec{v}_1 - \vec{v}_i, \vec{v}_{\infty A} = \vec{v}_2 - \vec{v}_f$$

- Transform  $\vec{v}_{\infty D}$  to ECI (Earth centered inertial) frame and then calculate the right ascension and declination to identify the co orbital transfer trajectory by using inclination ( $i$ ) of the orbit.
- Compute the impulsive  $\Delta V$  as follows:

1. For Fly-by mission

For the minimum  $\Delta V$  requirement, the parking orbit perigee and transfer hyperbola perigee should be same. Therefore the hyperbolic orbit perigee velocity is given by

$$v_h = \sqrt{\frac{2\mu_E}{r_p} + v_{\infty D}^2}, \quad (1)$$

But the velocity at the perigee of the departure orbit is given by

$$v_p = \sqrt{\frac{2\mu_E}{r_p} - \frac{2\mu_E}{r_p + r_a}}, \quad (2)$$

Hence the departure impulsive  $\Delta V_1$  is computed as

$$\Delta V_1 = v_h - v_p, \tag{3}$$

where  $\mu_E$  is the gravitational constant of central body Earth,  $r_p$  and  $r_a$  are the perigee altitude and apogee altitude of the parking orbit, respectively. For fly-by mission  $\Delta V = \Delta V_1$ .

2. For orbiter mission

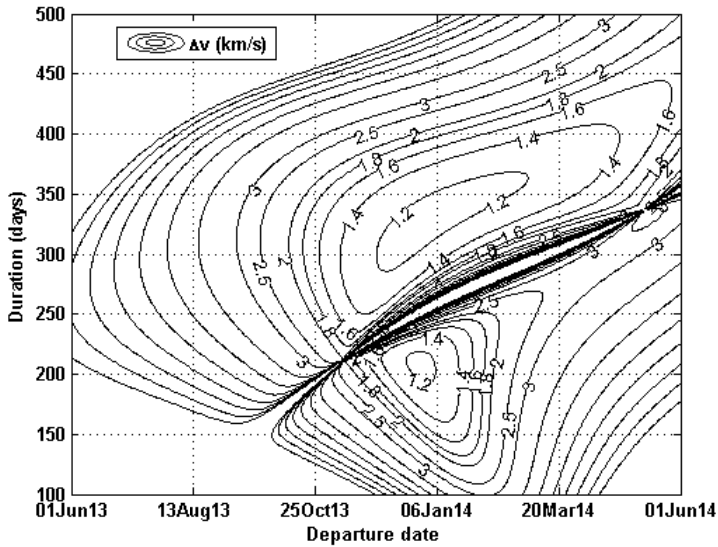
Here the impulsive  $\Delta V$  is given as follows

$$\Delta V = \Delta V_1 + \Delta V_2, \tag{4}$$

where  $\Delta V_1$  is the departure impulse calculated from equations (1)-(3) and  $\Delta V_2$  is the arrival impulse.  $\Delta V_2$  can be calculated by replacing  $\mu_E$ ,  $v_{\infty D}^2$ ,  $r_p$  and  $r_a$  with the suitable constraints at arrival phase of the planet Mars in equation (1) and (3).

### 5 Results and Discussions

In the present study, Earth to Mars transfer is considered from a GTO orbit of 180 x 36000 km with 65.5° inclination during the opportunity in 2013-2014. For finding out the minimum energy departure, we have considered the departure dates (DD) from 1 June 2013 to 1 June 2014 with the transfer duration ( $\Delta t$ ) of 100 to 500 days. Fig. 2 is the contour plot generated for departure impulsive  $\Delta V_1$ .



**Fig. 2.** Contour plot for direct ballistic transfer from Earth to Mars with parking orbit as GTO and 65.5° inclination

This figure is generated with the step size of one day in departure date as well as in the transfer duration. From the Fig. 1 it is clear that the minimum impulsive  $\Delta V_1$  exist between November 2013 and February 2014. Further, locating the minimum energy opportunity from the contour plot by varying the inclination ( $i$ ) and the step size for DD and  $\Delta t$  will substantially increase the computation time. So, the TEP algorithm is implemented to obtain the Minimum energy opportunities for direct ballistic transfer from Earth to Mars for both flyby and orbiter missions. The search space for TEP also involves inclination from  $0^0$  to  $90^0$ . In order to find the global minimum energy opportunity for flyby mission to Mars (case-1), minimization of departure impulsive  $\Delta V_1$  is considered as an objective function. TEP algorithm was coded in MATLAB and the simulations performed on a Pentium 4 computer with 2 GHz CPU under Windows XP. To evaluate the performance, the algorithm was run 50 times with 200 generations each, having 100 populations. The convergence plot for all the 50 runs is plotted in Fig. 3.

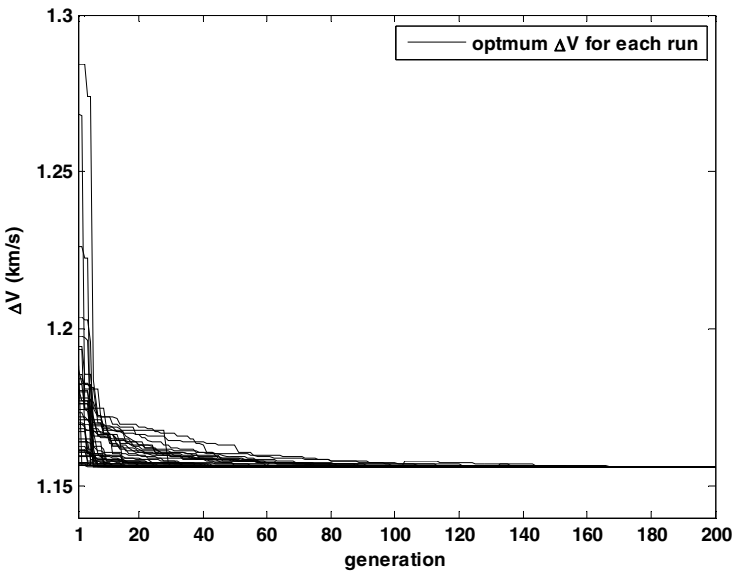


Fig. 3. The convergence plot of 50 runs in TEP over iterations (or generations)

Fig. 4 show the variation of optimum (or minimum) departure impulsive  $\Delta V_1$  obtained for each run using TEP. In order to find out the effectiveness of TEP, we have compared the results with GA also. The population size is selected to be 100. Because all optimization parameters have specified range, a binary coded GA is utilized and all parameters are coded in 60 bits. A single point crossover with probability of 0.8, and bit-wise mutation with probability of 0.01 are selected. Fig. 5 depicts the variation of optimum departure impulsive  $\Delta V_1$  obtained for each run using GA.

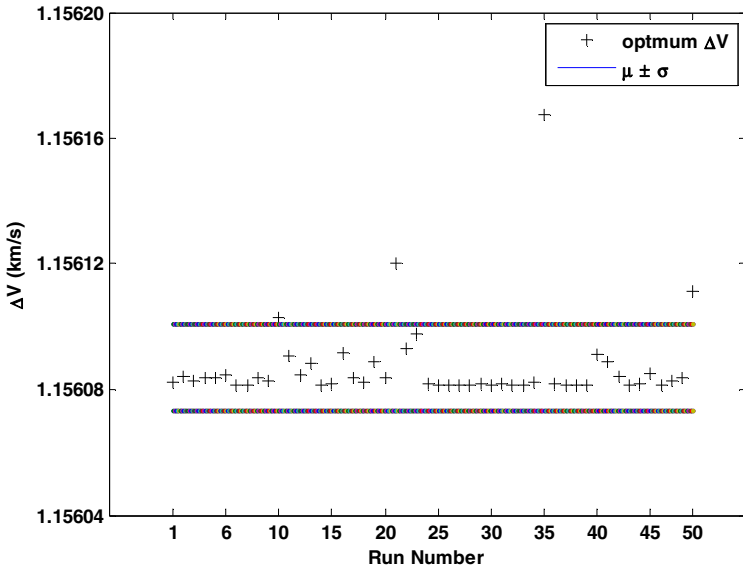


Fig. 4. The variation of optimum departure impulsive  $\Delta V_1$  obtained for each run using Taboo Evolutionary Programming

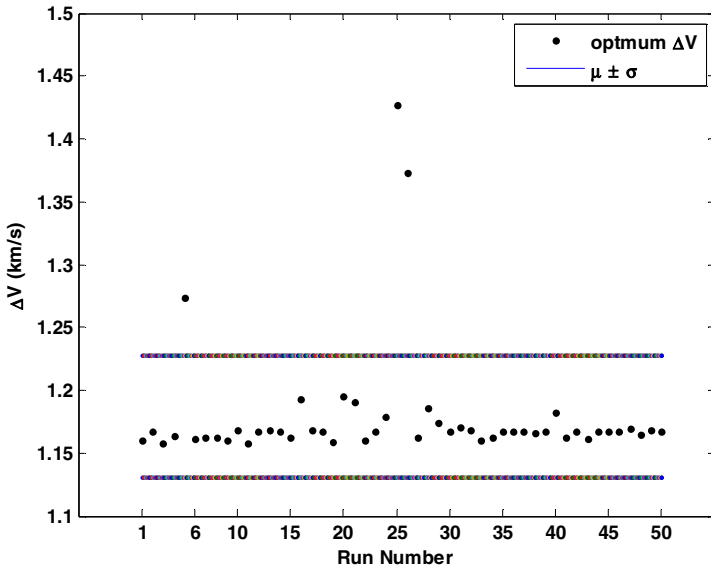


Fig. 5. The variation of optimum (or minimum) departure impulsive  $\Delta V_1$  obtained for each run using genetic algorithm (GA)

Comparing the Figs. 4 and 5, we can clearly understand that TEP perform better than GA. Two more test cases for orbiter missions with orbits 1000 x 1000 km (case-2) and 1000 x 80000 km (case-3) are considered around Mars. The total impulsive  $\Delta V$

calculated from equation (4) is minimized for orbiter mission. Table-1 shows the results obtained from 50 runs using optimization GA as well as TEP. From the Table, it is evident that the standard deviation ( $\sigma$ ) is very less in TEP comparing to GA for all cases. Similarly the difference between maximum and minimum values of  $\Delta V$  is less in TEP for all the cases. Also the same observation is valid in the difference between maximum and minimum replications performed to find optimum  $\Delta V$ . Table-2 provides the details of optimum velocity transfer details obtained from TEP for all three cases. The search space consists of GTO orbit (180 X 36000 km) as parking orbit and inclination 0 to 90 degrees in addition to DD and  $\Delta t$ .

Recently, an improved EP algorithm is published in [12]. It will be used to improve the present results in future.

**Table 1.** The dispersion measures for 50 runs

Optimum $\Delta V$ (km/s)						
	genetic algorithm (GA)			Taboo evolutionary programming (TEP)		
	Case-1	Case-2	Case-3	Case-1	Case-2	Case-3
Mean	1.173792	4.043652	2.847604	1.156087	3.497006	2.316875
$\sigma$	0.046200	0.738613	0.455012	0.000014	0	0
Minimum	1.156144	3.497229	2.317090	1.156081	3.497006	2.316875
Maximum	1.432042	8.593118	4.644549	1.156168	3.497006	2.316875
replications performed to reach optimum solution						
Minimum	51	90	87	100	80	81
Maximum	200	200	200	198	123	130

**Table 2.** Optimum impulsive  $\Delta V$  requirements using TEP

Case-1: Fly-by mission	
Minimum $\Delta V$ (km/s)	1.156081 km/s
Departure date duration (UTC)	From 2013 Dec 31 05:43:18.295 To 09:13:13.946
Transfer duration (days)	328.039 to 328.192
EPO inclination (deg)	65.99 to 65.83
Case-2: Orbiter mission with Martian orbit 1000X1000 km	
Minimum $\Delta V$ (km/s)	3.497006
Departure date (UTC)	04 Dec 2013 09:25:59.691
Transfer duration (days)	294.132
EPO inclination (deg)	88.66
Case-3: Orbiter mission with Martian orbit 1000X80000 km	
Minimum $\Delta V$ (km/s)	2.316875
Departure date (UTC)	04 Dec 2013 09:25:55.35
Transfer duration (days)	294.132
EPO inclination (deg)	88.67



## 6 Conclusions

In this paper we have studied to use the Taboo evolutionary programming based algorithm to solve the optimization problem in Earth to Mars transfer. Using this algorithm, we successfully identified the global minimum in three cases. In addition to this, the algorithm was also able to identify a launch opportunity which ensures the minimum  $\Delta V$ . Although the results presented in this paper are promising, a lot of work is to be done in the implementation of TEP for optimal multiple gravity assist trajectories.

## References

1. Yao, X., Liu, Y., Lin, G.: Evolutionary programming made faster. *IEEE Trans. Evol. Comput.* 3, 82–102 (1999)
2. Cvijovic, D., Klinowski, J.: Taboo search: an approach to the multiple minima problem. *Science* 267, 664–666 (1995)
3. Glover, F.: Tabu search- part I. *ORSA Journal of Computing* 1, 190–206 (1989)
4. Glover, F.: Tabu search-part II. *ORSA Journal of Computing* 2, 4–32 (1990)
5. Rajesh, J., Jayaraman, V.K., Kulkarni, B.D.: Taboo search algorithm for continuous function optimization. *Trans. IChemE* 78, Part A (2000)
6. Ji, M., Klinowski, J.: Taboo evolutionary programming: a new method of global optimization. *Proc. R. Soc. A* 462, 3613–3627 (2006)
7. Ji, M., Klinowski, J.: Convergence of taboo search in continuous global optimization. *Proc. R. Soc. A* 462, 2077–2084 (2006)
8. Ji, M., Tang, H., Guo, J.: A single-point mutation evolutionary programming. *Inf. Process. Lett.* 90, 293–299 (2004)
9. Minhazul Islam, S., Das, S., Ghosh, S., Roy, S., Suganthan, P.N.: An Adaptive Differential Evolution Algorithm with Novel Mutation and Crossover Strategies for Global Numerical Optimization. accepted by *IEEE Trans. on SMC-B* (2011)
10. Vallado, D.A.: *Fundamentals of astrodynamics and applications*, 2nd edn. Kluwer Academic Publishers, London (2001)
11. Battin, R.H.: *An introduction to the mathematics and methods of astrodynamics*. AIAA education series (1987)
12. Mallipeddi, R., Mallipeddi, S., Suganthan, P.N.: Ensemble strategies with adaptive evolutionary programming. *Information Sciences* 180(9), 1571–1581 (2010)

# Solving Redundancy Optimization Problem with a New Stochastic Algorithm

Chun-Xia Yang and Zhi-Hua Cui\*

Complex System and Computational Intelligence Laboratory,  
Taiyuan University of Science and Technology, Shanxi, 030024, China  
cuizhihua@gmail.com

**Abstract.** In order to solve the real-world problem which named Cleveland heart disease classification problem, we used a new stochastic optimization algorithm that simulate the plant growing process. It employs the photosynthesis operator and phototropism operator to mimic photosynthesis and phototropism phenomenons, we call it briefly with APPM algorithm. For the plant growing process, photosynthesis is a basic mechanism to provide the energy from sunshine, while phototropism is an important character to guide the growing direction. In this algorithm, each individual is called a branch, and the sampled points are regarded as the branch growing trajectory. Phototropism operator is designed to introduce the fitness function value, as well as phototropism operator is used to decide the growing direction. Up to date, there is little applications. Therefore, in this paper, APPM is successfully applied to the redundancy optimization problem. The objective of the redundancy allocation problem is to select from available components and to determine an optimal design configuration to maximize system reliability. BP neural network is trained to calculate the objective fitness, while APPM is applied to check the best choice of feasibility of solution. One example is used to illustrate the effectiveness of APPM.

## 1 Introduction

The redundancy problem is one important passport to enhance complicated system's dependability. Because the resources are often restricted, e.g. cost, volume, energy consumption et al., a key problem in the use of redundancy is how to allocate each part's redundant degrees. Generally, satisfying the objective's requests and reducing the needs of resources as far as possible are the main goals for the system's dependability. The method of redundant technique is to minimize the cost under the requesting of dependability[1][2][3][4][5]. Up to date, the main methods for reliability optimization are meta-heuristics[6][7][8] and exact algorithms[9][10]. Since exact algorithms can only provide high quality solutions for some special cases, most researchers pay their attentions to metaheuristic algorithms include GA[11][12], SA[13], and TS[14].

---

\* Corresponding author.

As a new population-based stochastic optimization stagy, APPM is proposed by Zhihua Cui et al. in 2011. It is a new stochastic algorithm to simulate the plant growing process. The total process is mapped into the optimization process, and each individual is called a branch, the search pattern is viewed as the branch growth process. To illustrate the effectiveness, it is applied to solve the directing orbits of chaotic systems. This paper will apply APPM to the optimization of reliability of a complex system.

In this paper, the stochastic simulation is introduced to provide a fitness value. Due to the amount computational time, neural network is used to approximate uncertain functions. Back-Propagation (BP) neural network[15] is a multi-layered feed-forward back-propagation network that is trained according to the err back-propagation algorithm.

The rest of this paper was organized as follows: the detailed description of redundant optimization problem is given in section 2, as well as in section 3, the details of APPM is presented. The proposed algorithm is discussed in section 4. Simulation results are listed in final section.

## 2 Problem Description

In a general system consisting of  $n$  stages, providing redundancy for components is perhaps the most common approach to design- for-reliability. During the system operation, all redundant elements operate simultaneously, only when the active element fails, one of the redundant elements begins to work. With this manner, there are two ways to provide redundancy for components: parallel redundancy and standby redundancy[16]. System's redundancy design is required to determine the optimal number of redundant elements for each component, and mainly has the following two kinds of questions: (1)how to optimize system reliability with the restriction of charge;(2) how to minimize the charge with the restriction of system reliability.

For a system consists of  $n$  components, suppose that every component has only one kind of elements to select, and it has only one way to provide redundancy: parallel redundancy or standby redundancy. Suppose the  $i^{th}$  components consist of  $x_i$  redundant elements ( $i = 1, 2, \dots, n$ ), the problem of redundancy optimization is

casting about for the optimal value of  $\vec{x} = (x_1, x_2, \dots, x_n)$  to optimize the system performance to be the best.

One key problem is how to estimate the system lifetime when the value of the vector  $\vec{x} = (x_1, x_2, \dots, x_n)$  is determined. For such a given decision vector  $\vec{x}$ , suppose that the redundant elements  $j$  operating in components  $i$  have lifetimes  $\xi_{ij}$ ,  $j = 1, 2, \dots, x_i, i = 1, 2, \dots, n$ , respectively, for convenience, we use the vector  $\vec{\xi} = (\xi_{11}, \xi_{12}, \dots, \xi_{1x_1}, \xi_{21}, \xi_{22}, \dots, \xi_{2x_2}, \dots, \xi_{n1}, \xi_{n2}, \dots, \xi_{2x_n})$  to denote the lifetimes of all redundant elements in the system.

In practice, the lifetime  $\xi$  of elements is usually a random vector. Thus the component lifetimes  $T(x, \xi)$  and system lifetime  $T_i(x, \xi)$  are also random variables for  $i = 1, 2, \dots, n$ . One of system performances is the expected lifetime  $E[T(x, \xi)]$ . It is obvious that the greater the expected lifetime  $E[T(x, \xi)]$ , the better the decision  $x$ .

### 3 APPM

In this paper, we only consider the following unconstrained problem:

$$\min f(x) \quad x \in [L, U]^D \subseteq R^D \tag{1}$$

#### 3.1 Main Method

To simulate the plant growth phenomenon, one important issue is to map this process into the optimization problem. Because light intensity guides the plant growing direction, and the photosynthesis provides necessary energy, the light intensity can be viewed as the fitness value which guides the search direction in the problem space. Furthermore, one point can be viewed as a branch, and the search strategy can be regarded as the growing trajectory. Because this new algorithm simulates the growing patten of plants by incorporating photosynthesis and phototropism mechanism, we call it briefly with APPM algorithm.

#### 3.2 Photosynthesis Operator

Photosynthesis is a process that converts carbon dioxide into organic compounds, especially sugars, using the energy from sunlight[17]. Photosynthesis occurs in plants, algae, and many species of bacteria, but not in archaea. Photosynthetic organisms are called photoautotrophs, since they can create their own food[18]. The rate of energy capture by photosynthesis is immense, approximately 100 terawatts[19], which is about six times larger than the power consumption of human civilization. As well as energy, photosynthesis is also the source of the carbon in all the organic compounds within organisms' bodies. In all, photosynthetic organisms convert around 100–115 teragrams of carbon into biomass per year[20].

Because the fitness value of each branch represents the light intensity, therefore, to avoid the confusion, a predefined range [0,1] is needed, in this paper, the following equation is designed to refine this area:

$$Score_u(t) = \frac{f_{worst}(t) - f(x_u(t))}{f_{worst}(t) - f_{best}(t)} \tag{2}$$

where  $f_{worst}(t)$  and  $f_{best}(t)$  are the worst and best original light intensity at time  $t$ , respectively,  $f(x_u(t))$  denotes the branch  $u$ 's original light intensity.

Photosynthetic rate plays an important role to measure how much energy produced. Up to date, many models have been proposed, such as rectangular hyperbolic model and non-rectangular hyperbolic model[21]. In this paper, rectangular hyperbolic model is employed to measure the obtained energy for each branch:

$$PR_u(t) = \frac{\alpha Score_u(t) P_{\max}}{\alpha Score_u(t) + P_{\max}} - R_d \quad (3)$$

where  $PR_u(t)$  represents the photosynthetic rate of branch  $u$  at time  $t$ ,  $Score_u(t)$  denotes the light intensity,  $\alpha$  is the initial quantum efficiency,  $P_{\max}$  is the maximum net photosynthesis rate, and  $R_d$  is the dark respiratory rate.  $\alpha$ ,  $P_{\max}$  and  $R_d$  are three parameters to control the size of photosynthetic rate. According to the corresponding references[21], they are set to 0.055, 30.2, 1.44, respectively.

In each iteration, all branches grow with obtained energy from photosynthesis according to Eq.(3).

### 3.3 Phototropism Operator

Phototropism is directional growth in which the direction of growth is determined by the direction of the light source. In other words, it is the growth and response to a light stimulus. Phototropism is most often observed in plants, but can also occur in other organisms such as fungi. The cells on the plant that are farthest from the light have a chemical called auxin that reacts when phototropism occurs. This causes the plant to have elongated cells on the farthest side from the light. Phototropism is one of the many plant tropisms or movements which respond to external stimuli. Growth towards a light source is a positive phototropism, while growth away from light is called negative phototropism (or Skotropism). Most plant shoots exhibit positive phototropism, while roots usually exhibit negative phototropism, although gravitropism may play a larger role in root behavior and growth. Some vine shoot tips exhibit negative phototropism, which allows them to grow towards dark, solid objects and climb them.

Because each branch will be attracted by those position with high light intensities, therefore, in iteration  $t$ , branch  $u$  takes the following movement:

$$x_u(t) = x_u(t-1) + Gp \cdot rand() \quad (4)$$

where  $Gp$  is a parameter reflecting the energy conversion rate and used to control the growing size per unit time.  $F_u(t)$  denotes the growing force guided by photosynthetic rate,  $rand()$  represents a random number sampled with uniformly distribution.

For each branch  $u$ ,  $F_u(t)$  is computed by

$$F_u(t) = \frac{F_u^{total}(t)}{\|F_u^{total}(t)\|} \quad (5)$$

where  $\| \cdot \|$  means the Euclidean distance,  $F_u^{total}(t)$  is computed as follows:

$$F_u^{total}(t) = \sum_{j \neq u} F_{u,j}(t) \tag{6}$$

and

$$F_{u,j}(t) \begin{cases} 0, \text{ if } \|x_u(t) - x_j(t)\| = 0, \\ coe \cdot \frac{e^{-\text{dim} \cdot PR_j(t)} - e^{-\text{dim} \cdot PR_u(t)}}{\|x_u(t) - x_j(t)\|}, \text{ otherwise.} \end{cases} \tag{7}$$

where  $\text{dim}$  represents the problem dimensionality,  $coe$  is a parameter used to control the movement direction:

$$coe = \begin{cases} 1, \text{ iff } (x_u(t)) > f(x_u(t)), \\ -1, \text{ otherwise.} \end{cases} \tag{8}$$

Furthermore, a small probability  $pm$  is introduced to reflect some random events affection:

$$x_u(t) = L + (U - L) \cdot rand_1(), \text{ if } (rand_2() < pm) \tag{9}$$

where  $L$  and  $U$  are the lower and upper bounds in the problem space,  $rand_1$  and  $rand_2$  are two random numbers with uniform distribution, respectively.

### 4 BP Neural Network

Artificial Neural Network (ANN) is basically as implied model of the biological neuron and uses an approach similar to human brain to make decisions and to arrive at conclusions[22].

The neuron can be defined as

$$y = f(W \times X + \theta_j) = f\left(\sum_{i=1}^n \omega_{ij} x_i - \theta_j\right) \tag{10}$$

where,  $x$  is input signals,  $\omega_{ij}$  is synaptic weights of neuron,  $f$  is the activation function and  $y$  is the output signal of neuron.

It consists of one input layer, one output layer and hidden layer. It may have one or more hidden layers. All layers are fully connected and of the feedforward type. The outputs are nonlinear function of inputs, and are controlled by weights that are computed during learning process[23].

The detail steps of Solving the redundancy optimization Problem are listed as follows :

**Step1.** Produce input and output data(namely training samples) or the uncertain functions by stochastic simulations.

**Step2.** Utilize the training samples to train BP neural network to approach the uncertain functions.

**Step3.** Initializing all individuals respectively. The initial position of individuals randomly distributed in  $[x_{\min}, x_{\max}]$ .

**Step4.** Update all individuals. For  $j^{th}$  individual, the  $\vec{x}_{best}(t)$  is updated according to Eq.(11). For all population, the  $\vec{Status}_{best}(t)$  is updated according to Eq.(9).

**Step5.** Computing the fitness value of each individual according to BP neural network.

**Step6.** Select the best individual which has the maximal fitness value.

**Step7.** If the iterative times are enough, output the best solution; otherwise, goto step 4.

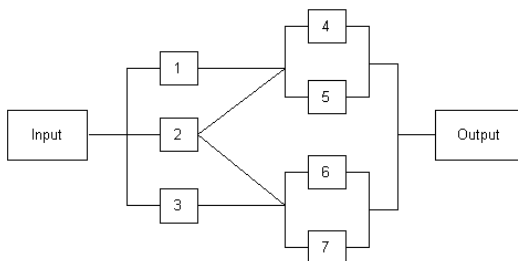
## 5 Simulation Results

Let us consider one life-support system in a space capsule[24] shown in Figure 1. For simplicity, we suppose that there is only one type of element to be selected for each Fig. 4. Using APPM to solve redundancy optimization Problem component, and all components are paralleled redundancy. The lifetimes of elements are assumed to be sampled with Gaussian distribution  $N(\mu, \sigma^2)$  shown in Tab.1.

The decision vector may be represented by  $\vec{x} = (x_1, x_2, \dots, x_7)$ , where  $x_j (j = 1, 2, \dots, 7)$  denotes the selected number of the  $j^{th}$  types of elements, respectively.

**Table 1.** Random Lifetimes and Prices of Elements

Type	1	2	3	4	5	6	7
Lifetime	$N(290, 21^2)$	$N(533, 23^2)$	$N(312, 25^2)$	$N(276, 23^2)$	$N(350, 26^2)$	$N(291, 21^2)$	$N(271, 24^2)$
Price	56	50	64	70	79	45	28



**Fig. 1.** Life-support System in a Space Capsule

The cost is computed by  $C(\vec{x}) = 56x_1 + 50x_2 + 64x_3 + 60x_4 + 79x_5 + 45x_6 + 28x_7$  from Figure 1. If the total capital available is 600, then we have a constraint  $C(\vec{x}) \leq 600$ . For the redundancy system, since we wish to maximize the expected lifetime  $E[T(\vec{x}, \xi)]$  subject to the cost constraint, we have the following stochastic expected lifetime maximization model:

$$\begin{cases} \max E[T(\vec{x}, \xi)] \\ s.t. \\ C(\vec{x}) \leq 600 \\ \vec{x} \geq 1 \end{cases} \quad (11)$$

In order to solve this stochastic expected lifetime maximization model of the life-support system in a space capsule, we deal with the uncertain function  $U: \vec{x} \rightarrow E[T(\vec{x}, \xi)]$  by stochastic simulation. Then a BP neural network including 7 input neurons, 12 hidden neurons and 1 output neuron is trained to approximate the uncertain function U[25].

The proposed social emotional optimization algorithm run 300 generations, where 5000 data are selected in BP training in which each simulation run 10000 iterations.

The obtained the cost is 596, the value  $E[T(\vec{x}, \xi)]$  is 375.14, and the optimal solution is  $\vec{x} = (1,1,1,1,3,1,3)$ .

## 6 Conclusion

This paper applies a new intelligent algorithm, APPM to solve the redundancy optimization problems. Simulation results show APPM is effective for the stochastic expected model problems. Future research topics includes the application of APPM to the other problems.

**Acknowledgments.** This paper were supported by National Natural Science Foundation of China under Grant 61003053, Natural Science Foundation of Shanxi Province under Grant 2011011012-1 and Shanxi Province Higher School’s Outstanding Young Academic Leaders Plan of China.

## References

1. Holland, J.H.: *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor (1975)
2. Kou, W., Prasad, V.R.: An annotated overview of system reliability optimization. *IEEE Transactions on Reliability* 49, 176–197 (2000)



3. Kou, W., Kim, T.: An overview of manufacturing yield and reliability modeling for semiconductor products. *Proc. IEEE* 87(8), 1329–1346 (1999)
4. Chern, M.S.: On the computational complexity of reliability redundancy allocation in a series system. *Operations Research Letters* 11, 309–315 (1992)
5. Misra, K.B.: On optimal reliability design: A review. *System Science* 12, 5–30 (1986)
6. Melachrinoudis, E., Min, H.: A tabu search heuristic for solving the multi-depot, multi-vehicle, double request dial-a-ride problem faced by a healthcare organisation. *Int. J. of Operational Research* 10(2), 214–239 (2011)
7. Kohda, T., Inoue, K.: A reliability optimization method for complex systems with the criterion of local optimality. *IEEE Trans. Reliability* R-31(1), 109–111 (1982)
8. Shahul Hamid Khan, B., Govindan, K.: A multi-objective simulated annealing algorithm for permutation flow shop scheduling problem. *Int. J. of Advanced Operations Management* 3(1), 88–100 (2011)
9. Baxter, L.A., Harche, F.: On the optimal assembly of series parallel systems. *Operations Research Letters* 11, 153–157 (1992)
10. Misra, K., Misra, V.: A procedure for solving general integer programming problems. *Microelectronics and Reliability* 34(1), 157–163 (1994)
11. Sathappan, O.L., Chitra, P., Venkatesh, P., Prabhu, M.: Modified genetic algorithm for multiobjective task scheduling on heterogeneous computing system. *Int. J. of Information Technology, Communications and Convergence* 1(2), 146–158 (2011)
12. Lu, J.-G., Li, H.-L., Chen, F.-X., Chen, L.: Combining strategy of genetic algorithm and particle swarm algorithm for optimum problem of RFID reader. *Int. J. of Innovative Computing and Applications* 3(2), 71–76 (2011)
13. Ravi, V., Murty, B., Reddy, P.: Nonequilibrium simulated annealing algorithm applied reliability optimization of complex systems. *IEEE Trans. Reliability* 46, 233–239 (1997)
14. Glover, F., Laguna, M.: *Tabu Search*. Kluwer Academic Publishers (1997)
15. Martin, H.T., Howard, D.B., Mark, B.H.: *Neural network design*. PWS Publishing, Boston (1996)
16. Bai, D.S., Yun, W.Y., Cheng, S.W.: Redundancy optimization of k-out-of-n:G systems with common-cause failures. *IEEE Trans. Reliability* 40, 56–59 (1991)
17. Smith, A.L.: *Oxford dictionary of biochemistry and molecular biology*. Oxford University Press, Wellington (1997)
18. Bryant, D.A., Frigaard, N.U.: Prokaryotic photosynthesis and phototrophy illuminated. *Trends in Microbiology* 14(11), 488–496 (2006)
19. Neelson, K.H., Conrad, P.G.: Life: past, present and future. *Philosophical Transactions of the Royal Society, Part B, Biological Sciences* 354(1392), 1923–1939 (1999)
20. Field, C.B., Behrenfeld, M.J., Randerson, J.T., Falkowski, P.: Primary production of the biosphere: integrating terrestrial and oceanic components. *Science* 281(5374), 237–240 (1998)
21. Piao, Y.Z., Qiang, Y.: Comparison of a new model of light response of photosynthesis with traditional models. *Journal of Shenyang Agricultural University* 38(6), 771–775 (2007)
22. Boryczka, M., Slowinski, R.: Derivation of optimal decision algorithms from decision ables using rough sets. *Bulletin of the Polish Academy of Sciences: Series Technical Sciences* 36, 252–260 (1988)
23. Ahn, B., Cho, S., Kim, C.: The integrated methodology of roughset theory and artificial neural-network for business failure prediction. *Expert Syst. Appl.* 18(2), 65–74 (2000)
24. Baxter, L.A., Harche, F.: On the optimal assembly of seriesparallel systems. *Operations Research Letters* 11 (1992)
25. Liu, B.: *Theory and practice of uncertain programming*, pp. 153–157. Springer-Verlag New York, LLC, New York (1992/2009)

# Energy Efficient Cluster Formation in Wireless Sensor Networks Using Cuckoo Search

Manian Dhivya<sup>1</sup>, Murugesan Sundarambal<sup>2</sup>, and J. Oswald Vincent<sup>3</sup>

<sup>1</sup> Department of Electrical and Electronics Engineering,  
Anna University of Technology,  
Coimbatore, TamilNadu-641047  
saidhivya1@gmail.com

<sup>2</sup> Department of Electrical and Electronics Engineering,  
Coimbatore Institute of Technology,  
Coimbatore, TamilNadu-641014  
mseecit@yahoo.co.in

<sup>3</sup> SHV Energy Private Limited,  
T.Nagar, Chennai, TamilNadu-600017  
oswaldrocks@gmail.com

**Abstract.** Wireless Sensor Networks consist of wide range of applications to be discerned and researched nowadays. The foremost restraint of these Networks is to reduce energy consumption and to prolong the lifetime of the network. In this paper a meta-heuristic optimization technique, Cuckoo Search is used to aggregate data in the Sensor Network. In the proposed technique, the least energy nodes are formed as subordinate chains (or) clusters for sensing the data and high energy nodes as Cluster Head for communicating to the base station. The Cuckoo search is proposed to get enhanced network performance incorporating balanced energy dissipation and results in the formation of optimum number of clusters and minimal energy consumption. The feasibility of the scheme is manifested by the Simulation results on comparison with the traditional methods.

**Keywords:** Wireless Sensor Networks, Clustering, Cuckoo Search, energy efficiency, Network Lifetime.

## 1 Introduction

Wireless Sensor Networks (WSNs) are distributed systems, limited in power, memory and computational capacities. The design often employs approaches such as energy-aware techniques, in-network processing, multihop communication and density control techniques to extend the network lifetime. While traditional networks aim to achieve high Quality of Service (QoS) provisions; Sensor Network protocols must focus primarily on power conservation [1]. Hence minimization and conservation of energy is a critical and significant issue in the design of Wireless Sensor Networks. Clustering and classification techniques afford a new dimension to the Sensor Network paradigm. The cluster based routing techniques are viable for a wide variety

of applications in WSN because of their divide and conquer strategy. It is widely accepted that balancing the energy dissipation among the nodes of the network is the key factor for prolonging the network lifetime [2]. Hence efficient data clustering techniques must be used to reduce the data redundancy and in turn reduce overhead on communication [3].

In this paper, Cuckoo Search [4] a metaheuristic approach is used for effective data collection. The least energy nodes are allowed to form subordinate chains or clusters and transmit the collected data to the Cluster-Head. The Cluster-Head (CH) is selected from the best fit of the search process. The CH transmits the aggregated data to the base station. Hence the least energy nodes are first exploited in communication, and periodically the search is done to rule out the inefficiencies of imbalance energy dissipation. The objective is to fairly balance the energy consumption among the sensor nodes, according to their residual energy and to extend the longevity of the network. The obtained results are compared with conventional methods to show the efficacy of the proposed method.

The rest of the paper is organized as follows. Theoretical Background is elaborated in section 2. The network model and the problem are formulated in section 3. Overview of Cuckoo Optimization, Proposed Cuckoo Search, and Methodology is discussed in section 4. Section 5 describes the simulation results. Finally conclusions are drawn in section 6.

## 2 Literature Overview

Several review articles, survey articles, and techniques are proposed for the past one decade on the energy conservation of WSNs. Low Energy Adaptive Clustering Hierarchy (LEACH) is a distributed single-hop clustering algorithm [5] proposed for energy utilization problem in Sensor Networks. The cluster head's role is periodically rotated among the sensor nodes to balance energy consumption. But cluster head rotation requires that, all the nodes be capable of performing data aggregation, cluster management and routing decisions. This results in extra hardware complexity in all the nodes. Hybrid Energy Efficient Distributed clustering (HEED) is one of the effective data-gathering protocols for Sensor Networks [6]. Both LEACH and HEED are applicable for mobile and static data collection.

Traditional cluster based routing have been extensively exploited. Hence an energy efficient routing protocol should encompass robustness, scalability, minimum overhead or delay, reduce data redundancy, multihop communication and shortest path routing. Therefore parallel solution methods are more desirable for fast Computation. In some duality models the network's optimization problem can be solved by a primal Parallel algorithm for allotted rates and a dual parallel algorithm for shadow prices or congestion signals and energy optimization [7]. The Computational Intelligence techniques and biologically inspired techniques can be integrated to get improved parallel solutions. The previous approaches have constraints in selecting shortest path which might not be a minimum energy cost route, decreasing the energy consumption

by replacing the hop-count routing with minimum energy routing and unpredictable node deaths. Hence Hybrid techniques and novel optimization techniques are utilized to compensate the deficiencies of one algorithm with another and to bring out cooperative performance.

Cuckoo Search is applied for both cluster formation as well as routing of gathered information to the base station. The research problem is divided into two perspectives, as follows: i) Cluster Formation phase and ii) Communication Phase from cluster head to base station. The traditional Cuckoo Search is modified as per the requirements of the proposed problem. To the best of the literature analysis done, this is the first paper incorporating Cuckoo Search technique for Wireless Sensor Networks.

### 3 Proposed Model

All nodes remain stationary and are initially charged with some base energy. Multi-hop situation is allowed for better communication link. Nodes can be arranged randomly in the two dimensional space. Constraints required for the base station from the nodes are neglected when the base station is located away from the network area. GPS devices which are used to sense the network nodes are neglected. Noise interference, signal fading and other losses are neglected during communication linkage. The distance between the 'n' sensors from the base station from the point P ( $x_i, y_i$ ) is given in (1).

$$d(i, j) = (x_i - x_n)^2 + (y_i - y_n)^2 . \quad (1)$$

$$E_{TX} = \{ l.Electrical + \epsilon_{fs}.d^2 (for 0 \leq d \leq d_{crossover} ) \} \quad (2)$$

(or)

$$E_{TX} = \{ l.Electrical + \epsilon_{mp}.d^4 (ford \geq d_{crossover} ) \} \quad (3)$$

The amount of energy consumed for transmission  $E_{Tx}$ , of  $l$ -bit message over a distance  $d$  is given in (2) and (3).

$$E_{RX} = l.Electrical \quad (4)$$

where  $Electrical = 50nJ/bit$  is the amount of energy consumed in electronic circuits,  $\epsilon_{fs} = 10pJ/bit/m^2$  is the energy consumed in an amplifier when transmitting at a distance shorter than  $d_{crossover}$ , and  $\epsilon_{mp} = 0.0013pJ/bit/m^4$  is the energy consumed in an amplifier when transmitting at a distance greater than  $d_{crossover}$ . The energy expended in receiving a  $l$ -bit message is given in (4). The assumptions for the Sensor Network are adopted from Aslam et al., [8] and the radio model is considered as stated in LEACH.

## 4 Cuckoo Search

The significance of optimization techniques had led them in the application of dynamic optimization problems like data aggregation and fusion, energy aware routing, task scheduling, security, optimal deployment and localization in Wireless Sensor Networks [9].

```

begin
Objective function  $f(x)$ ,  $x = (x_1, \dots, x_d)^T$ 
Generate initial population of
  n host nests  $x_i$  ( $i = 1, 2, \dots, n$ )
while ( $t < \text{MaxGeneration}$ ) or (stop criterion)
  Get a cuckoo randomly by Levy flights
  evaluate its quality/fitness  $F_i$ 
  Choose a nest among n (say, j) randomly
  if ( $F_i > F_j$ ),
    replace j by the new solution;
  end
  A fraction (pa) of worse nests are abandoned and
  new ones are built;
  Keep the best solutions (or nests with quality
  solutions);
  Rank the solutions and find the current best
end while
Postprocess results and visualization
end

```

Fig. 1. Pseudo Code for Cuckoo Search via Levy Flights

Cuckoo search (CS) is an optimization algorithm developed by Xin-She Yang and Suash Deb in 2009. It is a novel algorithm which is inspired by the obligate brood parasitism of some cuckoo species by laying their eggs in the nests of other host birds of other species. In the multi dimensional space where the optimal solution is sought, the CS is carried out for a maximization problem, where the quality or fitness of a solution can simply be proportional to the value of the objective function. Cuckoo Search has similarity to the hill climbing algorithm [10]. The CS is based on three idealized rules:

1. Each cuckoo lays one egg at a time, and dumps its egg in a randomly chosen nest.
2. The best nests with high quality of eggs will carry over to the next generation.
3. The number of available host's nests is fixed, and the egg laid by a cuckoo is discovered by the host bird with a probability  $P_a$ . In our experiment it is considered as 0.2 for 100 number of nests. The worst nests are discovered and dumped from further calculations.

The pseudo code for cuckoo Search is given in Figure 1. It can also be written in biased way with random step sizes. Equations (5), (6), and (7) give cuckoo search's biased random walk. Step size is determined as given below.

$$\text{Step size} = \text{rand} * (\text{nest}(\text{randperm}(n), :) - \text{nest}(\text{randperm}(n), :)); \quad (5)$$

$$\text{new\_nest} = \text{nest} + \text{stepsize} * K \quad (6)$$

$$\text{where; } K = \text{rand}(\text{size}(\text{nest})) > p_a \quad (7)$$

#### 4.1 Proposed Cuckoo Search

##### Step 1: Initialization

Select the number of sensor nodes, cuckoo nests, eggs in nests to start the search. Each nest has multiple eggs representing a set of solutions. Initialize the location and energy of nodes and the location of base station.

##### Step 2: Formation of Clusters using Cuckoo Search

$$f(df i) = \sum_{i=1}^{n-1} (100 * di) \quad (8)$$

The probability of choosing the best egg or quality egg is done by random walk. Step size and Levy angle is updated. In turn the nests are updated.

##### Step 3: Communication to the Base Station

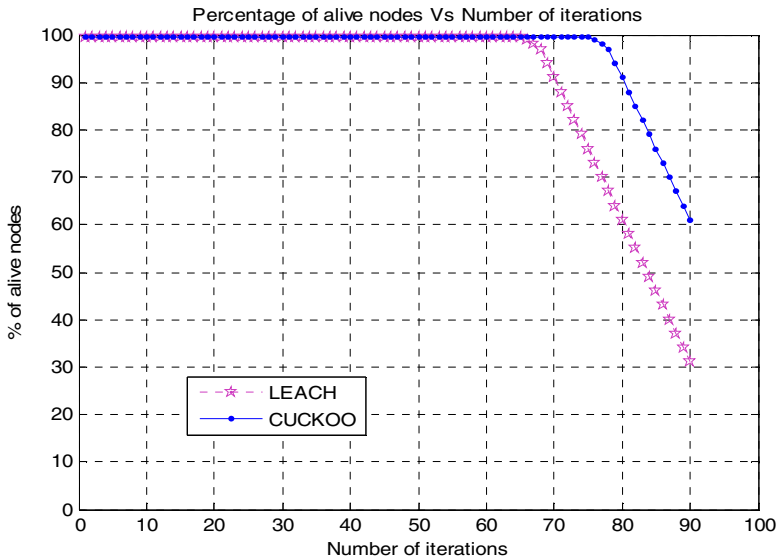
In Cuckoo Search the cuckoo tries to find the optimal path by minimum levy angle and random walk. Cuckoo traverses from a source node (Cluster) to the base station by travelling through neighbor clusters.

## 5 Simulation Results and Discussion

The simulations are carried out in MATLAB (7.11.0.584). A detailed survey and analysis of previous works is carried out and the simulation parameters are chosen in advance. The lifetime of the network is measured in iterations. The traditional methods HEED and LEACH are compared with the proposed scheme, with regard to the parameters relevant to network lifetime and energy consumption. Since the comparison models are LEACH and HEED, the network operation model similar to them is taken for analysis. In the analysis each round consists of a clustering phase followed by a data collection phase. Figure 2 shows the percentage of active nodes versus the number of iterations. In Table 1, the Simulation parameters are listed.

**Table 1.** Parameters

S.No	parameters	
1	Sensor deployment area	100 m *100m
2	Base station location	(50m,150m)
3	Number of nodes	100-200
4	Number of nests	100
5	Number of eggs in a nest	1-3
6	Data Packet size	100 bytes
7	Number of Rounds	100



**Fig. 2.** Network lifetime of 100 nodes vs. iterations

In Figure 3 the network lifetime is compared with regard to the iterations until the first node dies. The life time of a network is usually defined by the number of the nodes alive or percentage of nodes die. Figure 4 shows the number of rounds until the last nodes die versus number of nodes. The time the first node and last node dies are significant in determining the lifetime of the network. Cuckoo Search produces comparable results mainly due to search process in chain formation.

In practical environments, the technique is adaptive to communication and operation management, as the power consumption dominates a node’s power budget. Depending on the application, the power break-down and topology of the nodes are varied. The proposed work can be applied for any application entailing unequal energy distribution. For example, in Mobile object tracking, mobile source should track the designated target and retain target track information. Cuckoo Search can be applied for this scenario to form a cluster organization, and chase the object at ease by maintaining track route along a shortened path.

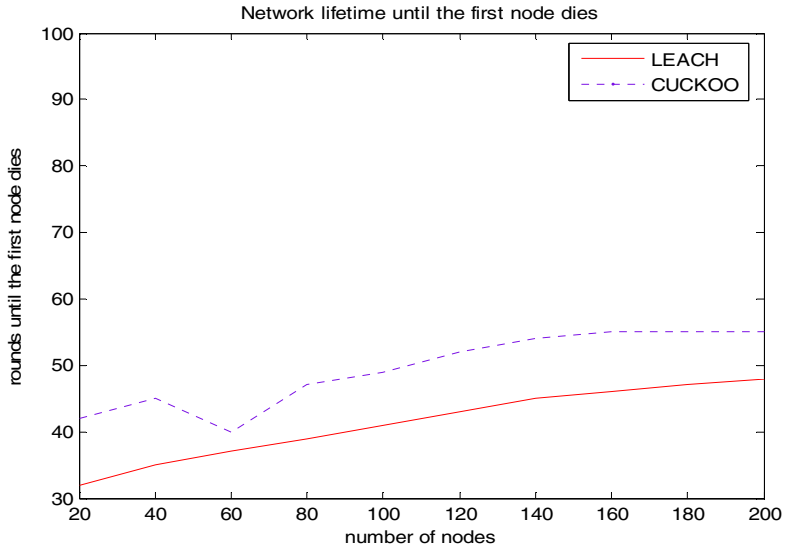


Fig. 3. Network lifetime until the first node dies

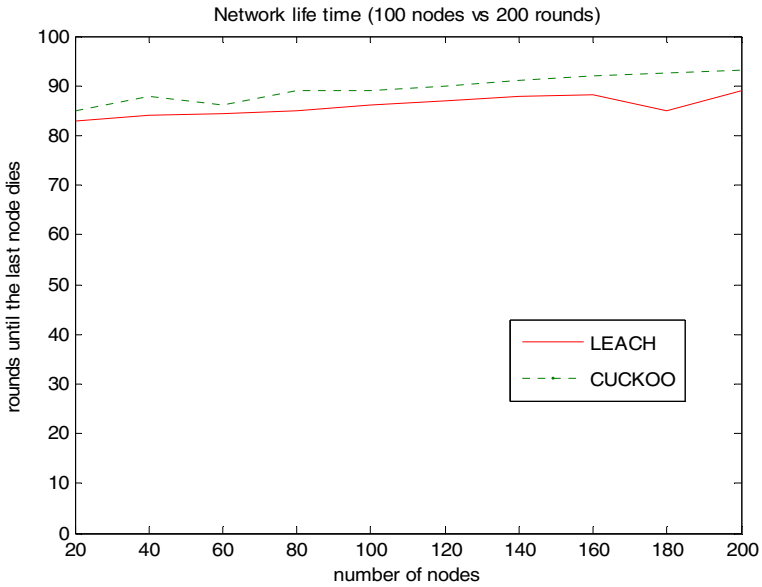


Fig. 4. Network lifetime until the last node dies



## 6 Conclusion

In this paper Cuckoo Search is applied for balancing the minimal energy dissipation among the Sensor nodes. All the nodes are utilized with equal importance to balance the energy dissipation. This approach incorporates two significant metrics that makes it favorable with respect to energy efficiency when compared to the original Cuckoo search. The proposed Cuckoo Search is compared with the standard LEACH protocol. The simulation results exhibits that, Cuckoo Search enhances the proportion of active nodes by minimum of fifteen percent. Future research will encompass the application of proposed Cuckoo for solving Swarm intelligence techniques combined with cross-layer design and Parameter.

## References

1. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless Sensor Networks: a survey. *Computer Networks* 38, 393–422 (2002)
2. Akojwar, A.G., Patrikar, R.M.: Improving Life Time of Wireless Sensor Networks Using Neural Network Based Classification Techniques with Cooperative Routing. *International Journal of Communications* 2(1), 75–86 (2008)
3. Chakraborty, W., Chakraborty, A., Mitra, S.K., Naskar, M.K.: An Energy Efficient scheme for data gathering in Wireless Sensor networks using Particle Swarm optimization. *Journal of Applied Computer Science* 6(3), 9–13 (2009)
4. Yang, X., Deb, S.: Cuckoo Search via Levy flights. Paper Presented at the Proc. of World Congress on Nature & Biologically Inspired Computing, (Nabic), pp. 210–214. IEEE, India (2009)
5. Heinzelman, W., Chandrakasan, A., Balakrishnan, H.: An application-specific Protocol architecture for wireless microsensor networks. *IEEE Transactions on Wireless Communications*, 660–670 (2002)
6. Younis, O., Fahmy, S.: HEED: a hybrid, energy-efficient, distributed Clustering approach for ad hoc sensor networks. *Transactions on Mobile Computing* 3, 660–669 (2004)
7. Sundarambal, M., Dhivya, M., Anbalagan, P.: Performance evaluation of Bandwidth Allocation in ATM networks. *Inderscience, International Journal of Business Information Systems* 6(3), 398–417 (2010)
8. Aslam, N., Phillips, W., Robertson, W.: A Unified Clustering and Communication Protocol for Wireless Sensor Networks. *IAENG International Journal of Computer Science* 35(3), [http://www.iaeng.org/IJCS/issues\\_v35/issue\\_3/IJCS\\_35\\_3\\_01.pdf](http://www.iaeng.org/IJCS/issues_v35/issue_3/IJCS_35_3_01.pdf)
9. Dhivya, M., Sundarambal, M., Anand, L.N.: Energy Efficient Computation Of Data Fusion in Wireless Sensor Networks Using Cuckoo Based Particle Approach (CBPA). *Int. J. Communications, Network and System Sciences* (April 2011), doi:10.4236/ijcns.2011.44030
10. Yang, X.-S., Deb, S.: Engineering optimization by cuckoo search. *Int. J. Mathematical Modelling and Numerical Optimization* 1(4), 330–343 (2010)

# Data Clustering Based on Teaching-Learning-Based Optimization

Suresh Chandra Satapathy<sup>1</sup> and Anima Naik<sup>2</sup>

<sup>1</sup> Anil Neerukonda Institute of Technology and Sciences,  
Vishakapatnam, India  
sureshsatapathy@ieee.org

<sup>2</sup> Majhighariani Institute of Technology and Sciences, Rayagada, India  
animanaik@gmail.com

**Abstract.** A new efficient optimization method, called ‘Teaching–Learning–Based Optimization (TLBO)’, has been proposed very recently for the optimization of mechanical design problems. This paper proposes a new approach to using TLBO to cluster data. It is shown how TLBO can be used to find the centroids of a user specified number of clusters. The new TLBO algorithms are evaluated on some datasets and compared to the performance of K-means and PSO clustering. Results show that TLBO clustering techniques have much potential.

**Keywords:** Clustering, Optimization, TLBO.

## 1 Introduction

Clustering algorithms can be grouped into two main classes of algorithms, namely supervised and unsupervised. This paper focuses on unsupervised clustering. Many unsupervised clustering algorithms have been developed. Most of these algorithms group data into clusters independent of the topology of input space. These algorithms include, among others, K-means [2, 3], ISODATA [4] and learning vector quantizers (LVQ)[5].

There have been many population based optimization techniques used for clustering in data mining literature. Among those PSO, DE, ACO, BF, ABC etc are widely used techniques. Recently a new efficient optimization method, called ‘Teaching–Learning–Based Optimization (TLBO)’[6], has been introduced for the optimization of mechanical design problems. This method works on the effect of influence of a teacher on learners. Like other nature-inspired algorithms, TLBO is also a population-based method and uses a population of solutions to proceed to the global solution. This paper explores the applicability of TLBO to cluster data vectors. The objective of the paper is to show that the TLBO algorithm can be used to cluster arbitrary data.

The rest of the paper is organized as follows: Section 2 presents an overview of the K-means algorithm; Section 3 presents overview of TLBO algorithm and in Section 4 the TLBO clustering technique is presented. PSO and PSO clustering are discussed in section 5. Experimental results are summarized in section 6. Section 7 gives the conclusion.

## 2 K-Means Clustering Algorithm

K-means clustering groups' data vectors into a predefined number of clusters, based on Euclidean distance as similarity measure. For the purpose of this paper, we define the symbols:  $N_d$  denotes the input dimension, i.e. the number of parameters of each data vector,  $N_0$  denotes the number of data vectors to be clustered,  $N_c$  denotes the number of cluster centroids (as provided by the user), i.e. the number of clusters to be formed,  $z_p$  denotes the p-th data vector,  $m_j$  denotes the centroid vector of cluster j,  $n_j$  is the number of data vectors in cluster j,  $C_j$  is the subset of data vectors that form cluster.

Using notation the standard K-means algorithm is summarized as

1. Randomly initialize the  $N_c$  cluster centroid vectors
2. Repeat
  - (a) For each data vector, assign the vector to the class with the closest centroid vector, where the distance to the centroid is determined using

$$d(z_p - m_j) = \sqrt{\sum_{k=1}^{N_d} (z_{pk} - m_{jk})^2} \quad (1)$$

where  $k$  subscripts the dimension.

- (b) Recalculate the cluster centroid vectors, using

$$m_j = \frac{1}{n_j} \sum_{z_p \in C_j} z_p \quad (2)$$

until a stopping criterion is satisfied.

The K-means clustering process can be stopped when either when the maximum number of iterations has been exceeded or when there is little change in the centroid vectors over a number of iterations or when there are no cluster membership changes. For the purposes of this study, the algorithm is stopped when a user-specified number of iterations have been exceeded a predefined value.

## 3 Teaching-Learning-Based Optimization

This optimization method is based on the effect of the influence of a teacher on the output of learners in a class. Like other nature-inspired algorithms, TLBO [6] is also a population based method that uses a population of solutions to proceed to the global solution. A group of learners are considered as the population. In TLBO, different subjects offered to learners are considered as different design variables for the TLBO. The learning results of a learner is analogous to the 'fitness', as in other population-based optimization techniques. The teacher is considered as the best solution obtained so far.

There are two parts in TLBO: 'Teacher Phase' and 'Learner Phase'. The 'Teacher Phase' means learning from the teacher and the 'Learner Phase' means learning through the interaction between learners.

### 3.1 Teacher Phase

In our society the best learner is mimicked as a teacher. The teacher tries to disseminate knowledge among learners, which will in turn increase the knowledge level of the whole class and help learners to get good marks or grades. So a teacher increases the mean learning value of the class according to his or her capability i.e. say the teacher  $T_1$  will try to move mean  $M_1$  towards their own level according to his or her capability, thereby increasing the learners' level to a new mean  $M_2$ . Teacher  $T_1$  will put maximum effort into teaching his or her students, but students will gain knowledge according to the quality of teaching delivered by a teacher and the quality of students present in the class. The quality of the students is judged from the mean value of the population. Teacher  $T_1$  puts effort in so as to increase the quality of the students from  $M_1$  to  $M_2$ , at which stage the students require a new teacher, of superior quality than themselves, i.e. in this case the new teacher is  $T_2$ .

Let  $M_i$  be the mean and  $T_i$  be the teacher at any iteration  $i$ .  $T_i$  will try to move mean  $M_i$  towards its own level, so now the new mean will be  $T_i$  designated as  $M_{new}$ . The solution is updated according to the difference between the existing and the new mean given by

$$Difference\_mean_i = r_i(M_{new} - T_F M_i) \quad (3)$$

where  $T_F$  is a teaching factor that decides the value of mean to be changed, and  $r_i$  is a random number in the range  $[0, 1]$ . The value of  $T_F$  can be either 1 or 2, which is again a heuristic step and decided randomly with equal probability as

$$T_F = round[1 + rand(0,1) * (2 - 1)] \quad (4)$$

This difference modifies the existing solution according to the following expression

$$X_{new,i} = X_{old,i} + Difference\_mean_i \quad (5)$$

### 3.2 Learner Phase

Learners increase their knowledge by two different means: one through input from the teacher and the other through interaction between themselves. A learner interacts randomly with other learners with the help of group discussions, presentations, formal communications, etc. A learner learns something new if the other learner has more knowledge than him or her. Learner modification is expressed as

For  $i = 1: P_n$

Randomly select two learners  $X_i$  and  $X_j$ , where  $i \neq j$

If  $f(X_i) < f(X_j)$   $X_{new,i} = X_{old,i} + r_i (X_i - X_j)$

Else  $X_{new,i} = X_{old,i} + r_i (X_j - X_i)$

End If

End For

Accept  $X_{new}$  if it gives a better function value.

## 4 TLBO Clustering

In the context of clustering, a single particle represents the  $N_c$  cluster centroid vectors. That is, each particle  $x_i$ , is constructed as follows:

$$x_i = (m_{i,1}, m_{i,2}, \dots, m_{i,j}, \dots, m_{i,N_c}) \quad (11)$$

where  $m_{i,j}$  refers to the  $j$ -th cluster centroid vector of the  $i$ -th particle in cluster  $C_{ij}$ . Therefore, a swarm represents a number of candidate clustering for the current data vectors. The fitness of particle is easily measured as the quantization error,

$$J_e = \frac{\sum_j^{N_c} [\sum_{\forall z_p \in C_{ij}} d(z_p, m_j) / |C_{ij}|]}{N_c} \quad (12)$$

where  $d$  is defined in equation (1), and  $|C_{ij}|$  is the number of data vectors belonging to cluster  $C_{ij}$  i.e. the frequency of that cluster.

The following section first presents a standard *TLBO* algorithm for clustering data into a given number of clusters.

### 4.1 TLBO Algorithm for Clustering

Using TLBO data vectors can be clustered as follows:

1. Initialize each learner to contain  $N$ , randomly selected cluster centroids.
2. For  $t = 1$  to  $t_{max}$  do
  - (a) For each learner  $i$  do
  - (b) For each data vector  $z_p$ 
    - i. Calculate the Euclidean distance  $d(z_p, m_{ij})$  to all cluster centroids  $C_{ij}$ .
    - ii. Assign  $z_p$  to cluster  $C_{ij}$ . such that  $(z_p, m_{ij}) = \min_{\forall c = 1, 2, \dots, N_c} d(z_p, m_{ic})$ .
    - iii. Calculate the fitness using equation (12)
  - (c) Update the learner modification
  - (d) Update the cluster centroids using equations (3) and (5).

where  $t_{max}$ , is the maximum number of iteration

## 5 Particle Swarm Optimization

Particle swarm optimization (PSO) is a population-based stochastic search process, modeled after the social behavior of bird flock [7, 8]. The algorithm maintains a population of particles, where each particle represents a potential solution to an optimization problem. In the context of PSO, a swarm refers to a number of potential solutions to the optimization problem, where each potential solution is referred to as a particle. The aim of the PSO is to find the particle position that results in the best evaluation of a given fitness (objective) function. Each particle represents a position in  $N_d$  dimensional space, and is "flown" through this multi-dimensional search space,

adjusting its position toward both the particle's best position found thus far and the best position in the neighborhood of that particle. Each particle  $i$  maintains information:  $x_i$  : The *current position* of the particle,  $v_i$  : The *current velocity* of the particle,  $y_i$  : The *personal best position* of the particle

Using notation a particle's position is adjusted according to

$$v_{i,k}(t + 1) = wv_{i,k}(t) + c_1r_{1,k}(t)(y_{i,k}(t) - x_{i,k}(t)) + c_2r_{2,k}(t)(y_k(t) - x_{i,k}(t)) \quad (13)$$

$$x_i(t + 1) = x_i(t) + v_i(t + 1) \quad (14)$$

where  $w$  is the inertia weight,  $c_1$  and  $c_2$  are the acceleration constants,  $r_{1,j}(t), r_{2,j}(t) \sim U(0,1)$  and  $K=1,2,\dots,N_d$ . The velocity is thus calculated based on three contributions:

- (1) a fraction of the previous velocity,
- (2) the cognitive component which is a function of the distance of the particle from its personal best position, and
- (3) the social component which is a function of the distance of the particle from the best particle found thus far (i.e. the best of the personal bests).

The personal best position of particle  $i$  is calculated as

$$y_i(t + 1) = \begin{cases} y_i(t) & \text{if } f(x_i(t + 1)) \geq f(y_i(t)) \\ x_i(t + 1) & \text{if } f(x_i(t + 1)) < f(y_i(t)) \end{cases} \quad (15)$$

Two basic approaches to PSO exist based on the interpretation of the neighborhood of particles. Equation (13) reflects the *gbest* version of PSO where, for each particle, the neighborhood is simply the entire swarm. The social component then causes particles to be drawn toward the best particle in the swam. In the *lbest* PSO model, the swam is divided into overlapping neighborhoods, and the best particle of each neighborhood is determined. For the *lbest* PSO model, the social component of equation (13) changes to

$$c_2r_{2,k}(t)(y_{j,k}(t) - x_{i,k}(t)) \quad (16)$$

where  $y_j$  is the best particle in the neighborhood of the  $i$ -th particle. The PSO is usually executed with repeated application of equations (12) and (13) until a specified number of iterations has been exceeded. Alternatively, the algorithm can be terminated when the velocity updates are close to zero over a number of iterations.

### 5.1 PSO Clustering

In [1] is the first paper to use PSO for clustering. In the context of clustering, a single particle represents the  $N_c$  cluster centroid vectors. That is, each particle  $x_i$ , is constructed as follows:

$$x_i (m_{i,1}, m_{i,2}, \dots, m_{i,j}, \dots, m_{i,N_c}) \quad (17)$$

where  $m_{i,j}$  refers to the  $j$ -th cluster centroid vector of the  $i$ -th particle in cluster  $C_{ij}$ . Therefore, a swarm represents a number of candidate clustering for the current data vectors. The fitness of particles are easily measured as the quantization error given in equation (12).

## 5.2 gbest PSO Clustering Algorithm

Using PSO data vectors can be clustered as follows:

1. Initialize each learner to contain  $N$ , randomly selected cluster centroids.
2. For  $t = 1$  to  $t_{max}$  do
  - (a) For each learner  $i$  do
  - (b) For each data vector  $z_p$ 
    - i. Calculate the Euclidean distance  $d(z_p, m_{ij})$  to all cluster centroids  $C_{ij}$ .
    - ii. Assign  $z_p$  to cluster  $C_{ij}$ . such that  $(z_p, m_{ij}) = \min_{\forall c = 1, 2, \dots, N_c} d(z_p, m_{ic})$ .
    - iii. Calculate the fitness using equation (12)
  - (c) Update the cluster centroids using equations (13) and (14).

where  $t_{max}$  is the maximum number of iteration.

## 6 Experimental Results

This section compares the results of the K-means, PSO Clustering and TLBO Clustering algorithms on five real world dataset and one artificial dataset. Iris, Glass, WBC, Wine, and Haberman's Survival dataset are taken from UCI machine repository. Artificial data set ( $n=385$   $d=2$ ,  $K=4$ ) consists of 385 data vectors, 2 features and 4 classes some data sets. The artificial dataset is designed manually and the Fig 1 represents the four clusters of artificial dataset.

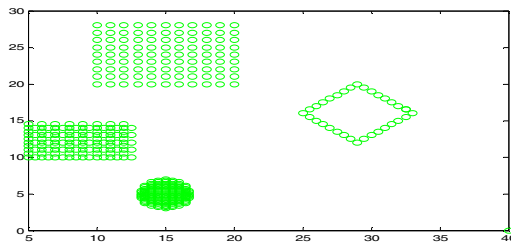


Fig. 1. Artificial data sets

The main purpose is to compare the quality of the respective clustering's, where quality is measured according to the following three criteria:

- the quantization error as defined in equation (12).
- the intra-cluster distances, i.e. mean of maximum distance between two data vectors within a cluster of clusters i.e.  $\frac{1}{N_c} \sum_{i=1}^{N_c} [\max_{z_p, z_q \in C_i} d(z_p, z_q)]$ , where the objective is to minimize the intra-cluster distances.

- the inter-cluster distances, i.e. minimum distance between the centroids of the clusters, where the objective is to maximize the distance between clusters.

The latter two objectives respectively correspond to crisp, compact clusters that are well separated.

For all the results reported, averages over 20 simulations are given. All algorithms are run for 1000 function evaluations, and the TLBO and PSO algorithms used 10 particles each in the simulations. For both the TLBO and PSO Clustering algorithm, we randomly initialize cluster centroids. The cluster centroids are also randomly fixed between  $X_{max}$  and  $X_{min}$ , which denote the maximum and minimum numerical values of any feature of the data set under test, respectively.

In this paper, while comparing the performance of algorithms, we focus on computational time required to find the solution. For comparing the speed of the algorithms, the first thing we require is a fair time measurement. The number of iterations or generations cannot be accepted as a time measure since the algorithms perform different amount of works in their inner loops, and they have different population sizes. Hence, we choose the number of *fitness function evaluations (FEs)* as a measure of computation time instead of generations or iterations. Since the algorithms are stochastic in nature, the results of two successive runs usually do not match. Hence, we have taken 20 independent runs (with different seeds of the random number generator) of each algorithm. The results have been stated in terms of the mean values and standard deviations over the 20 runs in each case. All the experiment codes are implemented in MATLAB. The experiments are conducted on a Pentium 4, 1GB memory desktop in Windows XP 2002 environment.

Table 1 summarizes the results obtained from the three clustering algorithms. The values reported are averages over 20 simulations, with standard deviations to indicate the range of values to which the algorithms converge. For all the problems, except for wine data, the TLBO algorithm had the smallest average quantization error. For the wine data, the K-Means clustering algorithm has a better quantization error.

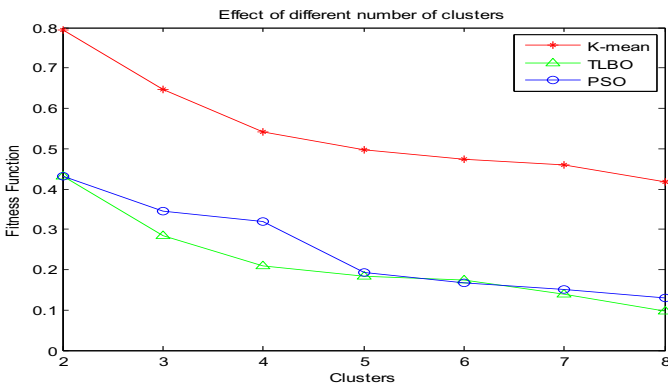
Due to space limitations we have summarized the results of IRIS dataset only in Fig 2 and 3. In Figure 2 the effect of varying number of clusters versus quantization error for Iris dataset is presented. As expected the quantization error goes down with increase in the number of clusters.

Figure 3 illustrates the convergence behavior of the algorithms for the IRIS data sets. The K-means algorithm exhibited a faster, but premature convergence to a large quantization error, while the TLBO algorithms had slower convergence, but to lower quantization errors. The K-means algorithm converged after 15 function evaluations, PSO algorithm converged after 90 evaluations and TLBO algorithm converge after 130 function evolution. In fact it can be clearly seen from the Fig 3 that after about 20 function evaluation the TLBO starts to stabilize. It clearly indicates the superiority of the TLBO over other two clustering approaches.



**Table 1.** Final Solution (Mean and Standard Deviation Over 20 Independent Runs)

Data sets name	Algorithm Used	Quantization error	Intra cluster distance	Inter cluster distance
<b>Iris data</b>	K-Means	0.6505±0.0124	2.5147±0.0064	1.7928±0.0045
	PSO	0.3023±0.0263	3.2806±0.2314	1.5403±0.1693
	TLBO	<b>0.2845±0.0047</b>	3.1350±0.2187	1.6939±0.2173
<b>Wine data</b>	K-Means	<b>101.9587±4.1863</b>	458.6488 ±3.6020	327.0094±58.2466
	PSO	342.2986±9.5387	456.6758±4.8859	279.7534±56.4985
	TLBO	345.0649±20.7124	450.7811±23.9000	278.1714±29.5149
<b>Breast cancer data</b>	K-Means	5.2318±0.0016	18.5776±0.00	13.8972±0.0060
	PSO	0.4719±0.2333	20.6651±0.00	13.4433±0.00
	TLBO	<b>0.0514±0.0171</b>	20.6651±0.00	13.4433±0.00
<b>Haberman's Survival Data Set</b>	K-Means	8.5926±0.0199	50.8811±2.7782	17.5622±0.3637
	PSO	6.8734±0.7443	50.6146±2.2313	17.6723±3.8535
	TLBO	<b>6.1354±0.3461</b>	50.2354±1.5903	16.2889±2.8530
<b>Glass data</b>	K-Means	0.8912±0.0156	3.8371±1.5920	3.6852±1.5902
	PSO	0.0574±0.0037	6.6325±0.8561	0.5290±0.2149
	TLBO	<b>0.0507±0.0106</b>	5.9905±0.6239	0.4924±0.1411
<b>Artificial data</b>	K-Means	3.3170±0.3378	12.5611±0.8066	9.5948±1.6502
	PSO	2.3735±0.2451	16.113±2.6502	4.7780±2.3012
	TLBO	<b>2.0437±0.1203</b>	16.5537±0.0523	5.0143±2.1310

**Fig. 2.** Effect of different number of clusters on Iris data sets

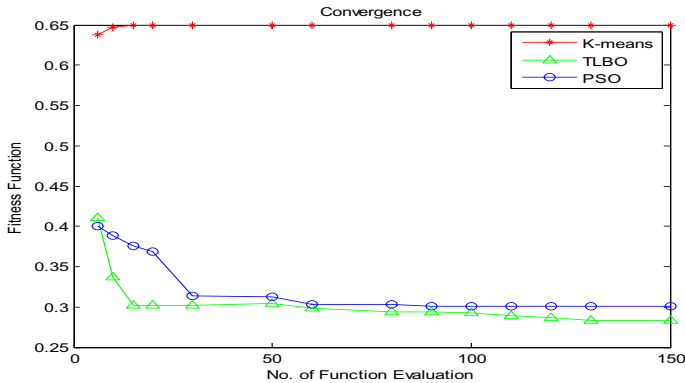


Fig. 3. Algorithm convergence for Iris data sets

## 7 Conclusion

This paper investigated the application of a new optimization algorithm known as teaching learning based optimization (TLBO) to cluster data vectors. The TLBO approach was compared against classical K-means clustering and PSO clustering. From the simulation results it is observed that TLBO may have a slow convergence but it has stable convergence trend much earlier compared to other two algorithms and better clustering results. As further study some parameter tuning of TLBO can be done to improve the convergence characteristics.

## References

1. van der Merwe, D.W., Engelbrecht, A.P.: Data Clustering using Particle Swarm Optimization. *IEEE Evolutionary Computation* 1, 215–220 (2003), doi:10.1109/CEC.2003.1299577
2. Forgy, E.: Cluster Analysis of Multivariate Data, Efficiency versus Interpretability of Classification. *Biometrics* 2, 768–769 (1965)
3. Hartigan, J.A.: *Clustering Algorithms*. John Wiley EL Sons, New York (1975)
4. Ball, G., Hall, D.: A Clustering Technique for Summarizing Multivariate Data. *Behavioral Science* 12, 153–155 (1967)
5. Fausett, L.V.: *Fundamentals of Neural Networks*. Prentice Hall (1994)
6. Rao, R.V., Savsani, V.J., Vakharia, D.P.: Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems. *Computer-Aided Design* 43, 303–315 (2011)
7. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In: *Proceedings of the IEEE International Joint Conference on Neural Networks*, vol. 4, pp. 1942–1948 (1995)
8. Kennedy, J., Eberhart, R.C., Shi, Y.: *Swarm Intelligence*. Morgan Kaufmann (2002)

# Extracting Semantically Similar Frequent Patterns Using Ontologies

S. Vasavi\*, S. Jayaprada, and V. Srinivasa Rao

Computer Science & Engineering Department,  
VRSiddhartha Engineering College (Autonomous),  
Affiliated to JNTU Kakinada, KANURU, Vijayawada, Krishna (DT),  
Andhra Pradesh, India  
vasavi\_movva@vrsiddhartha.ac.in

**Abstract.** Many methods were proposed to generate a large number of association rules efficiently. These methods are dependent on non-semantic information such as support, confidence. Also work on pattern analysis has been focused on frequent patterns, sequential patterns, closed patterns. Identifying semantic information and extracting semantically similar frequent patterns helps to interpret the meanings of the pattern and to further explore them at different levels of abstraction. This paper makes a study on existing semantic similarity measures and proposes a new measure for calculating semantic similarity using domain dependent and domain independent ontologies. This paper also proposes an algorithm SSFPOA (Semantically Similar Frequent Patterns extraction using Ontology Algorithm) for extracting and clustering semantically similar frequent patterns. The case study which is illustrated in this paper shows that the algorithm can be used to produce association rules at high level of abstraction.

**Keywords:** Frequent patterns, Association rules, Semantic similarity, Ontology, Clustering.

## 1 Introduction

Many methods were proposed for mining association rules efficiently [1-8] which consider measures such as support and confidence for measuring quality of the rules. Alternatively we can determine semantic similarity amongst frequent pattern items of a particular domain in order to generate related association rules. In this paper a study is made on the methods for inferring similarity within frequent patterns. From the patterns thus identified, we can extract the transactions which contain these patterns and cluster them, there by each cluster containing semantically similar patterns, which can be further explored to find related association rules. [9, 10] can be used to generate frequent patterns. Section 2 presents the background behind leading to the current problem. Section 3 presents the metric for calculating semantic similarity using domain dependent and domain independent ontologies that was proposed by us in [16] and its usage in the proposed algorithm. Section 4 presents conclusion and future work.

---

\* Corresponding author.

## 2 Related Work

The work given in [11] is to discover the hidden meanings of a frequent pattern by annotating it with in-depth, concise, and structured information. They proposed a general approach to generate such an annotation for a frequent pattern by constructing its context model, selecting informative context indicators, and extracting representative transactions and semantically similar patterns. It also incorporates user prior knowledge for discovering semantic relations. They consider only frequent item as context unit in finding semantic similarity, where as we consider structural information (the entire path items leading to the frequent pattern item) within the domain dependent ontology. The work given in [12] finds semantic similarity based on the notion of information content. Authors believe that the shorter the path from one node to another, the more similar they are. But shortest path is not only sufficient for confirming on similarity; semantic similarity between elements within the entire path should also be considered. Also their work is purely dependent on domain independent ontologies such as Wordnet[13]. Their work is purely on word similarity rather than context similarity. Problem with using wordnet is that words which are not captured by it are treated as noise. We first find morphological root word with the help of Web feature such as wordnet[13] and use NLP techniques such as tokenization, lemmatization, elimination and string based techniques such as n-gram are used during preprocessing. "Sugato Basu et al" [14] proposes a method of estimating the novelty of rules using WordNet. Semantic distance between two words is based on the length of the shortest path connecting  $w_i$  and  $w_j$  in WordNet. Average value of  $d(w_i; w_j)$  across all pairs of words  $(w_i; w_j)$  where  $w_i$  is in the antecedent and  $w_j$  is in the consequent of the rule is used. Our measure constructs semantic similarity matrix between all pairs of frequent items by considering not only domain dependent ontology (such as structural information) but also domain independent ontology (such as wordnet) and cluster all frequent patterns items with high similarity. These clusters are further used to generate association rules with high level abstraction. Our work is similar to [15] which finds semantic similarity between keywords using ontology and the statistical information like support, confidence, chi-squared value but differs in using domain knowledge across 12 matchers in 4 phases. They build ontology to describe the concepts and relationships in the domain for measuring the semantic relation and calculating the semantic similarity. Our proposed algorithm as explained in section III not only considers the depth of the node within the tree but also considers other structural features such as

1. The number of the children for each node of schema
2. The number of subclasses for each class for ontology.

### 2.1 Algorithm SSFPOA

SSFPOA uses 4 phases (12 matchers) and in each phase multiple matchers result is aggregated and send to the next phase.

1. Inputs to Pre-processing step:
  - (i) Isolated characteristics of elements in mapping pair, e.g. length of elements, number of tokens within each pattern item

- (ii) Syntactic characteristics of mapping pair eg. number of common tokens in each frequent pattern, tokens which are important to be matched
  - (iii) Domain independent ontology (Web feature) such as wordnet[13] is used to find morphological root word
  - (iv) NLP techniques such as tokenization, lemmatization, elimination.
2. After preprocessing tokens of the frequent patterns are given as input to name matcher for exact name matching
  3. Inputs to Linguistic matcher: Here 3 matchers such as Synonym, soundex and string matching techniques such as N-gram (trigram) are considered.
  4. Structural matcher step: Structural features such as:
    - (i) The number of the children for each node of schema
    - (ii) The depth of each node from the root
    - (iii) We consider number of common parents in the hierarchy (not at the same level), depth of the element, to conclude on equivalently less general (el), equivalently more general (em), equal (=), disjoint (dj) for leaf nodes and less general (LG), more general (MG) for non leaf nodes and further take user feedback to finally resolve el,em to = or dj
  5. Constraint check step(Cardinality check): This step identifies 1:1 or 1:n cardinality amongst item names for forming clusters. Auxiliary information such as domain dependent ontology (data dictionary), previous mappings (to determine transitive dependencies EX:  $a \rightarrow b$ ,  $b \rightarrow c$ , then  $a \rightarrow c$ ) and user feedback (to resolve el,em) is also considered.

Algorithm is outlined as follows:

```

i,j , l1,l2 node;
logicalrelation nodeList[][];
m,n int;
List=Check_Cluster(S1,S2)
If List==null continue else
If Find_k(S1,S2) == null then continue else
Findlist(S1,S2,Sk)
for each node j in S1 and S2 do
Tokenization(j)
Stemming(j)
Stopwords (j)
Abbreviations(j)
For each item l1 in S1 find path
storepath(l1)
For each item l2 in S2 find path
storepath(l2)
for each node i in S1 find match with every element j of S2 do
if i,j in Findlist(S1,S2,Sk) then continue else
logicalrelationij =flood(S1i, S2j ) //linguistic matcher//
For each item i of S1 //cardinality check//
For each item j of S2
Check for simple mappings (1:1)
Check for Complex mappings (1:n)
cluster(logicalrelationij ) // store j as synonym of i in synonymn file//

```

## 2.2 Case Study

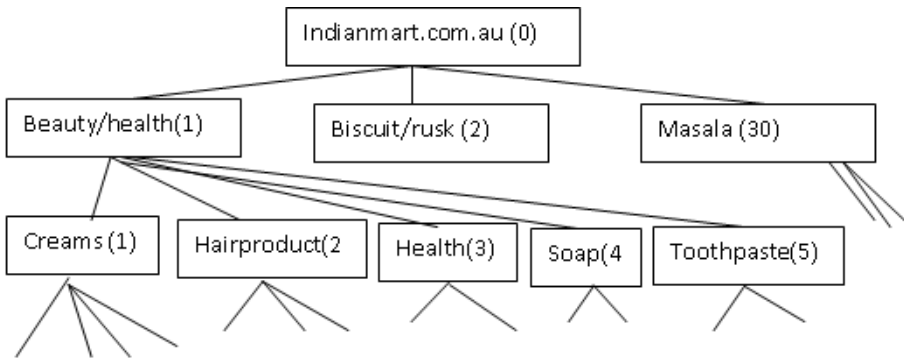
Figure 1 shows a part of the domain dependent ontology available at <http://www.indianmart.com.au/shopping/index.php> which has listed 625 products from different manufacturers (forming an ontology with 36 classes - 2 non leaf and 623 leafs, one more node is added as a virtual root).

- indianmart.com.au
- Beauty/Health
  - + Creams
  - + Hair products
  - + Health
  - + Soap
  - + Toothpaste
- + Biscuit/rusk
- + Dal/pulses
- + Desert/sweets
- + drinks
- + Flour
- + Frozen
  - + Paneer
  - + Roti/parata
  - + snacks
  - + Vegetables
- + Miscellaneous
- + Oil
- + Papad/Khakra
- + Phone cards
- + Pickle
- + Poojaitems/Incense
- + Rice
- + Sauce
- + Snacks/Namkeen
- + special
- + Spice/ powder/ herbs
- + Sweets
- + Tea/Coffee
- + Vegitable
- + Yogurt
- + Paste
- + Chutney
- + Seasoning/mix
- + Masala

**Fig. 1.** Ontology for super market items

Each class has three properties such as product image, item name, price in \$. We took one month transactions which came to 88162 transactions. We downloaded Apriori code available at [17] and made changes to it. [17] uses 2 input files config.txt (line 1 specifies number of items per transaction , line 2 mentions number of transactions, line 3 gives minimum support, while transa.txt - transaction file, each line is a transaction, items are separated by a space. But it requires more computation time as it involves integer arithmetic. We used encoding similar to [18] for transaction items which is described as follows:

Root is at level 0, its children are given BFS numbering 1,2,3,4 etc. Each of its children is in turn given BFS numbering etc as shown in figure 2.



**Fig. 2.** Encoding of transaction items

In order to specify list of items we took multidimensional integer array. This process of encoding helps for easy search of an item within the domain dependent ontology ( figure 1). Also minimum support count 2 (1764 transactions) is taken. Nearly 15,000 rules were produced. These rules were given as input to SSFPOA. (i.e) given a frequent pattern (rule)  $\{X,Y,Z\} \rightarrow N1$  and  $\{A,B,C\} \rightarrow N2$  where X,Y,Z,A,B,C are item names (in encoding format) and N1 and N2 are supports, SSFPOA finds semantic similarity among these items , clusters them, generates an association rules at high level abstraction as  $\{P,Q,R\} \rightarrow N1+N2$  where P,Q,R are the important tokens identified at preprocessing level. Semantic matrix which comes to 15,000X 15,000 elements is shown here. Table 1 shows some list of items which are semantically similar.

**Table 1.** Semantically similar product names

SNO	Semantically similar items	Parent node
1	Vicco vajardanti 100gm 3.75\$ Vicco vajardanti 100gm 6.50\$	Tooth paste
2	Gori Gori blue fairness bleach 50gm \$4.95 Gori Gori fairness bleach 50gm \$4.95	Creams
3	Godrej renew hair color cream black 120ml \$6.95 Godrej renew hair color cream brown 120ml \$6.95	Hair products

**Table 1.** (Continued)

4	Supreme Dark brown henna 150gm \$2.95 Supreme Maroon henna 150gm \$2.95	Hair products
5	Eno lemon 100gm \$2.95 Eno regular 100gm \$2.95	Health
6	Hajmola imli 120 tbs \$3.95 Hajmola regular 120 tbs	Health
7	Godrej No.1 natural \$2.75 Godrej No.1 rose \$2.75	Soaps
8	Neem active toothpaste 125gm \$3.25 Neem toothpaste 125gm \$3.25	Tooth paste
9	Pattu Jeera khari biscuit - 200gm \$2.95 Pattu Masala khari biscuit - 200gm \$2.95 Pattu plain khari biscuit - 200gm \$2.95 Pattu Tomato khari biscuit - 200gm \$2.95	Biscuit
10	Indian Mart Black Urid 1kg \$3.75 Indian Mart Black Urid split 1kg \$3.75	Dal/pulses
11	Indian Mart Kabuli Channa (10mm) 1kg \$3.75 Indian Mart Kabuli Channa (9mm) 1kg \$3.75	Dal/pulses
12	Indian Mart Toor Daal 1kg \$3.75 Indian Mart Toor Daal premium 1kg \$3.75	Dal/pulses
13	Katoomba Roti Parantha (20 roti) \$7.25 Katoomba Roti Parantha lite(20 roti) \$7.25	Roti
14	Mezban Gajar Halwa 280gms \$6.25 Mezban Loki Halwa 280gms \$6.25	Snacks
15	Taj Valor Lilva (indian beans) 400gm \$2.00 Taj Valor Papdi (indian beans) 400gm \$2.00	Vegetables
16	Food Color - Orange \$2.00 Food Color - Red \$2.00 Food Color - Yellow \$2.00	Miscellaneous
17	Aithra sago papad 200gm \$2.25 Aithra sago papad color 200gm \$2.25	Papad
18	Lijjat papad - garlic 200gm \$2.00 Lijjat papad -red chilli 200gm \$2.00 Lijjat papad - Urad 200gm \$2.00	Papad
19	South Asia Phone Card \$8.00 South Asia Phone Card \$16.00 South Asia Phone Card \$40.00	Phone cards
20	Priya garlic pickle \$2.50 Priya garlic pickle \$6.50	Pickles
21	Priya mixed vegetable pickle \$2.50 Priya mixed vegetable pickle \$6.50	Pickles
22	Priya red chilli pickle \$2.50 Priya red chilli pickle \$6.50	Pickles
23	Aithra pepper 100gm \$3.50 Aithra pepper 100gm \$1.95	Spice/herbs/powder
24	Shan bihari kabab BBQ mix \$1.75 Shan bihari kabab BBQ mix \$1.75	Seasoning/mix



Table 2 gives some of items for which EM (equally more general) values are generated.

**Table 2.** Equally more general product names

SNO	Equally more general	Remarks
1	Complan Chocolate 500gms \$8.95 Complan Vanilla 500gms \$8.50	Cost is different
2	Supreme Black henna 6 pouches \$3.95 Supreme Dark brown henna 150gm \$2.95	6 pouches (each of 25gm) equals to 150 gm
3	Ayers Rock Roti Parantha 20 pcs 1.3Kg \$7.25 Katoomba Roti Parantha (20 roti) 1.3Kg \$7.25	Roti/Parata Roti/Parata
4	Ayers Rock Shredded cocunut 400gm \$2.75 Badshah Shredded cocunut 400gm \$2.75	Vegetables

Table 3 list some of the items with disjoint as output as parents are different (but are similar).

**Table 3.** Disjoint product names

SNO	Equally more general	Parent node
1	Haldiram Chum Chum 1kg \$6.50 Haldiram Chum Chum 1kg \$5.95	Sweets Desserts/sweets
2	Haldiram Rasogula 1kg \$6.50 Haldiram Rasogula 1kg \$5.95	Sweets Desserts/sweets
3	Haldiram Soan Papdi 500g \$6.50 Haldiram Soan Papdi 500g \$6.50	Sweets Desserts/sweets
4	Maharaja Kala Janum 1kg \$6.50 Maharaja Gulab Jamun 1kg \$6.50 Maharaja Kala Janum 1kg \$5.95	Sweets Desserts/sweets
5	Maharaja Shahi Jamun 1kg \$6.50 Maharaja Choice Shahi Jamun 1kg \$5.95	Sweets Desserts/sweets

Table 4 gives some items with equally less general as they come from different manufacturers.

**Table 4.** Equally less general product names

SNO	Equally less general	Parent node
1	Cadbury Bournvita 500gms \$8.95, Complan Chocolate 500gms \$8.95	Health
2	Mother's Recipe cut mango pickle \$3.75 Priya cut mango pickle \$6.50	Pickle
3	Ayers Rock Roti Parantha 20 pcs 1.3Kg \$7.25 Katoomba Roti Parantha (20 roti) 1.3Kg \$7.25	Roti/Parata
4	Ayers Rock Shredded cocunut 400gm \$2.75 Badshah Shredded cocunut 400gm \$2.75	Vegetables

### 3 Conclusions and Future Work

Our algorithm SSFPOA has focused on interpreting the frequent patterns that are mined, especially extracting semantically similar items and clustering them. Initially we took product catalogue of B2B trade. Our future work concentrates on domains such as text mining (where author and co-author are semantically same), taxonomies related to biological categories such as gene synonyms where similar genes can probably be replaced. Our future work will also focus on estimating the performance of each individual matcher, ie Linguistic Matcher (after 5<sup>th</sup> Matcher, and 8<sup>th</sup> matcher ) structural matcher (after 9<sup>th</sup> matcher), constraint matcher (after 12<sup>th</sup> matcher), Does propagation of similarity value after each matcher improving or not, Does any other extra matchers are to be used such as edit distance, prefix, suffix (at preprocessing level) or at element Level (such as to prove that cat and dog are equal as they both come from the ancestors pets). How does our algorithm perform w.r.t quality measures such as precision and recall and performance measurement such as time required for producing result, Number of Proof Steps for mapping should be considered.

### References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association Trules. In: Proc. 20th Int'l Conf. on Very Large Databases, pp. 487–499 (1994)
2. Toivonen, H.: Sampling large databases for association rules. In: Int'l Conf. on Very Large Databases, pp. 134–145 (1996)
3. Srikant, R., Vu, Q., Agrawal, R.: Mining association rules with item constraints. In: Proc. of the 3rd Int'l Conf. on KDD and Data Mining (KDD 1997), Newport Beach, California (August 1997)
4. Yang, G., Shimada, K., Mabu, S., Hirasawa, K., Hu, J.: Mining Equalized Association Rules from Multi Concept Layers of Ontology Using Genetic Network Programming. In: Proc. of IEEE Cong. on Evolutionary Computation (CEC 2007), Singapore, pp. 705–712 (September 2007)
5. Yang, G., Shimada, K., Mabu, S., Hirasawa, K., Hu, J.: A Genetic Network Progrmming Based Method to Mine Generalized Association Rules with Ontology. Journal of Advanced Computational Intelligence and Intelligent Informatics (2006)
6. Mannila, H., Toivonen, H., Verkamo, A.I.: Efficient algorithms for discovering association rules. In: Proc. of the AAAI Workshop on Knowledge Discovery in Databases, pp. 181–192 (July 1994)
7. Shimada, K., Hirasawa, K., Hu, J.: Class Association Rule Mining with Chi-Squared Test Using Genetic Network Programming. In: Proc. of IEEE Int'l Conf. on Systems, Man and Cybernetics (ICSMC 2006), pp. 5338–5344 (October 2006)
8. Shimada, K., Hirasawa, K., Hu, J.: Genetic Network Programming with Acquisition Mechanisms of Association Rules. Journal of Advanced Computational Intelligence and Intelligent Informatics 10(1), 102–111 (2006)
9. Agrawal, R., Imieliski, T., Swami, A.: Mining association rules between sets of items in large databases. In: Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, pp. 207–216 (1993)

10. Han, J., Pei, J., Yin, Y., Mao, R.: Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data Min. Knowl. Discov.* 8(1), 53–87 (2004)
11. Mei, Q., Xin, D., Cheng, H., Han, J., Xiang, C.: *Semantic Annotation of Frequent Patterns* (2007)
12. Resnik, P.: *Using information content to evaluate semantic similarity in a taxonomy* (1995)
13. <http://www.wordnet.princeton.edu>
14. Basu, S., Mooney, R.J., Pasupuleti, K.V., Ghosh, J.: Evaluating the Novelty of TextMined Rules Using Lexical Knowledge. In: *Proceedings of the Seventh International Conference on Knowledge Discovery and Data Mining* (2001)
15. Yang, G., Shimada, K., Mabu, S., Hirasawa, K.: *A Personalized Association Rule Ranking Method Based on Semantic Similarity And Evolutionary Computation* (2008)
16. Vasavi, S.: *Semantic interoperability within heterogeneous environments using schema matching*. PhD Thesis (2010)
17. Magnus, N., Hamilton, H.: *package apriori*, Copyright: University of Regina, Nathan Magnus and Su Yibin (June 2009)
18. Han, J., Fu, Y.: Mining Multiple-Level Association Rules in Large Databases. *IEEE Transactions on Knowledge and Data Engineering* 11(5) (September/October 1999)


# Correlating Binding Site Residues of the Protein and Ligand Features to Its Functionality

B. Ravindra Reddy<sup>1</sup>, T. Sobha Rani<sup>1</sup>, S. Durga Bhavani<sup>1</sup>,  
Raju S. Bapi<sup>1</sup>, and G. Narahari Sastry<sup>2</sup>

<sup>1</sup> Computational Intelligence Lab,  
Department of Computer and Information Sciences,  
University of Hyderabad, Hyderabad, India

<sup>2</sup> Molecular Modeling Group,  
Indian Institute of Chemical Technology, Hyderabad, India

**Abstract.** Machine learning tools are employed to establish relationship between the characteristics of protein-ligand binding site and enzyme class. Enzyme classification is a challenging problem from data mining perspective due to (i) class imbalance problem and (ii) appropriate feature selection. We address the problem by choosing novel features from protein binding site. Protein Ligand Interaction Database (PLID), which gives a comprehensive view of binding sites in a protein along with other contact information, is updated and presented here as PLID v1.1. The database facilitates the study of protein-ligand interaction. Novel features due to protein ligand interaction including the chemical compound features as well as fraction of contact and tightness are investigated for classification task. The weighted classification accuracy for the data set with binding site residues as features is found to be 56% using a Random Forest classifier. It may be concluded that either the binding site features are not adequately representing the enzyme class information or the problem is caused due to the class imbalance. This problem needs further investigation.

**Availability**  PLID v1.1 is publicly available at <http://dcis.uohyd.ernet.in/~plid> and mirrored at <http://203.199.182.73/gnsmmg/databases/plid>

## 1 Introduction

One of the important objectives of modern biology is to understand and predict protein function from its sequence and structure. Existing methods are based on transferring the annotation from a homologous protein but this method fails when a similar protein with reliable annotation cannot be identified. Even predicting function of a protein where structure is obtained through a close homologue has misleading annotations [Bork *et al.*, 1998, Devos *et al.*, 2000, Rost *et al.*, 2003]. While the earlier methods are based on sequence information, the excellent progress made in the 3D-structure characterization provides

<sup>1</sup> Contact: [sdbcs@uohyd.ernet.in](mailto:sdbcs@uohyd.ernet.in); [gnsastry@iiict.res.in](mailto:gnsastry@iiict.res.in).

more valuable information on the important regions of proteins. Obviously, the protein sites which are amenable for ligand complexation are functionally the most important. Number of structures with annotation 'Unknown Function' deposited in the Protein Data Bank (PDB) [PDB] has nearly trebled each year during 1999 and 2004 [Dobson *et al.*, 2005], [Watson *et al.*, 2004].

The problem of protein function prediction is addressed in a top down approach by [Shen *et al.*, 2007]. They classify a given protein as belonging to one of the top Enzyme Commission (EC) classes, viz. oxidoreductases, transferases, hydrolases, lyases, isomerases and ligases and achieve a remarkable performance accuracy of over 90% for EC class prediction using only sequence data approach. Here features of the order of a few thousands are employed.

The question that is being addressed in this work is about contribution of binding site features to the functionality of an enzyme. One of the road blocks to this study is the non-availability of adequate number of experimentally determined protein structures. Machine learning algorithms perform well on large data as shown by the sequence based approaches which obtain very good performance accuracy of about 90% [Shen *et al.*, 2007] for enzyme classification problem. On the other hand, Bray *et al.* [Bray *et al.*, 2009] who use a non-redundant set of 294 enzymes predict an enzyme class with an accuracy of 33.1%. Clearly, using structures to predict function is a challenging problem both from data perspective as well as features perspective. Related work by Bray *et al.* [Bray *et al.*, 2009] and Dobson *et al.* [Dobson *et al.*, 2005] extract structural features from binding site available in Catalytic Site Atlas (CSA) and analyze differences in sequence and structural features between the six main EC classification groups.

The focus of this paper is on determining function from using features of binding site alone. We develop a prediction method that can perform EC classification utilizing features extracted from the protein-ligand co-crystals. Random forest classifier achieves an overall weighted accuracy of 56%. The study addresses the importance of binding site residues together with the ligand molecule in guiding the ligand binding process for determining protein function.

**Literature on Databases:** Specialized databases like ZINC and PLID provide information regarding chemical features of the ligand molecules as well as protein-ligand co-complex interaction information [Irwin *et al.*, 2005], [Reddy *et al.*, 2008]. Protein structure information is obtained from PDB [PDB]. In this paper we report the updation of our database, PLID v1.0 which comprises the description of binding sites of all protein-ligand complexes available in PDB upto January 2011.

## 1.1 ZINC Database

ZINC [Irwin *et al.*, 2005] is a free database of commercially available chemical compounds for virtual screening including drug-like compounds. ZINC contains a library of nearly 750,000 molecules, each with 3D structure and are annotated with molecular properties. Currently ZINC provides 9 calculated properties - molecular weight, logP, De\_apolar, De\_polar, number of HBA, number of HBD,

tPSA, charge and NRB for each molecule. Some of the features computed from SMILES notation obtained from ZINC database are number of atoms of Carbon, Oxygen and Nitrogen; number of non-metal atoms of Florine and Sulphur; nature of cyclic/acyclic information, chirality etc.

## 1.2 Protein Ligand Interaction Database (PLID v1.1)

Protein Ligand Interaction Database (PLID v1.0) built by [Reddy et al., 2008](#) was developed from PDB available upto 2006 with a comprehensive view of each active site in the protein and the other contact information that is available in the database to facilitate the study of protein ligand interactions. PLID consists of binding area residues within 4Å for all the complexed proteins in PDB along with physico-chemical, thermal and quantum chemical properties of the ligands and the binding site. Here it also quantifies the interaction strength. As the PDB grows in size continuously every year, we undertook the task of updating the database and publish the next version called PLID v1.1. Among the 44,144 proteins present in the PDB till September 2008, about 74% of the proteins i.e., 32,522 are found to be bound with at least one ligand molecule. The current work uses the latest data available till January 2011 but the updation of PLID is in progress.

## 2 Can Binding Site Analysis Predict Protein Functional Classes?

How much the binding site features contribute to the functionality of the enzyme? As compared to the sequence based studies which obtain very good performance accuracy of about 90% [Shen et al., 2007](#), this study focuses on determining function from using features of binding site alone.

Having obtained all the protein ligand complexes and grouped them into different EC classes, our attempt will be to explore correlation between binding site and the functional class of protein. So far in the literature catalytic site residues have been considered for protein classification [Bray et al., 2009](#). Catalytic site residues are obtained from Catalytic Site Atlas (CSA), a resource of catalytic sites and residues identified in enzymes using structural data [Porter et al., 2004](#). Bray et al. achieve classification accuracy of 33.1%. The main focus of this paper is to enhance the understanding of relation between the structure of an enzyme to its function analyzing the sequence and structural features of the six enzyme classes. They have also considered many features like secondary structure content (helices etc.) and surface area and so on in their analysis apart from the amino acid content.

### 2.1 Data Preparation

Many of the structures in PDB are redundant due to several reasons and one of them is that a protein gets deposited more than once. The repetition mainly

happens when a protein is studied under different experimental conditions or with different ligands or inhibitors. We build a non-redundant protein set from the initial set of proteins. Using the four levels of EC numbers, proteins having identical least significant EC number are grouped together. From each group, proteins with more than 90% sequence identity are removed. These steps are repeated for the proteins at next significant EC digit and so on up to the top EC class number. The final set will now contain proteins having less than 90% sequence identity with other members. Thus a distinct set of proteins within each class are obtained on which further analysis steps could be made.

**Removal of Non-Ligands.** Distinguishing non-ligands from ligands cannot be easily automated. We assume that ligands are those having very close contact with protein. The property of fraction of contact ( $f$ ) proves to be useful in distinguishing ligands from non-ligands. Figure 2 shows how the profile of fraction of contact varies across the functional classes. We extract those binding sites which have ( $FC$ ) value greater than 0.4 or if the number of protein residues in contact is more than 4. This step will remove almost all non-ligands.

To summarize, the following is the sequence of the the Data Preparation Steps.

**Step-1:** Removing proteins having more than 90% sequence identity

**Step-2:** Removing of non-ligand molecules

**Step-3:** Removing instances of identical binding sites

**Step-4:** Removing protein complexes having prosthetic and bulkier groups

Final data set after the four steps of data preparation is given in Table 1.

**Table 1.** Various protein classes and their class sizes

Protein class	Size of class
Hydrolase	743
Isomerase	120
Ligase	88
Lyase	158
Oxidoreductase	415
Transferase	686

### 3 Feature Extraction

Classification experiments are carried out on enzymes obtained after the data cleaning process. These enzymes belong to six classes, namely, oxidoreductase, transferase, hydrolase, lyase, isomerase and ligases.

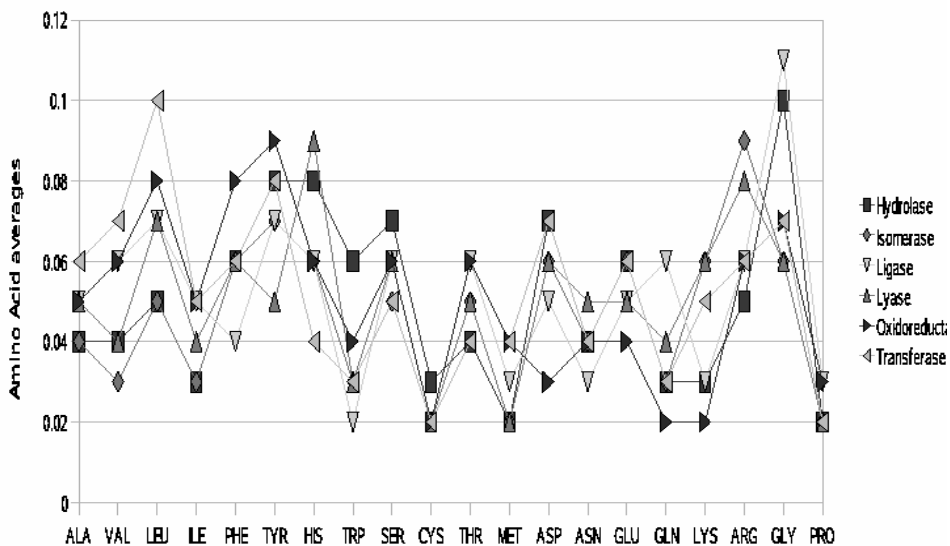


Fig. 1. Amino acid composition across the six enzyme classes is shown

### 3.1 Enzyme Features

Amino acid compositions are used as features to represent the binding site residues. The distribution of amino acid composition at binding sites among the selected protein classes is shown in Figure 1.

The uneven distribution of amino acid occurrences over the protein classes can be observed from Figure 1. For example, Leu level is observed to be highest for transferases than other classes. Similarly Gly is highest for Ligases and His for Lyases. Clearly AA composition seems to delineate protein function classes and this is an important characteristic required by a good feature to be used for classification purposes. The abundance of aromatic residues and glycine in the active site was also observed by [Soga et al., 2007](#) wherein they analyzed the binding pockets of drug-like compounds on the surface of proteins in the PDB. [Malik et al., 2007](#) similarly observed abundance of aromatic residues in carbohydrate binding proteins.

### 3.2 Ligand Features

Modules Ligand Extractor and BERF (Binding Environment Residue Finder) of PLID identify binding residues and calculates two important properties such as fraction of contact ( $FC$ ) and average tightness ( $T$ ) which quantify the interaction between the protein and ligand as described in [Reddy et al., 2008](#). Many other ligand features like count of atoms C, O, N, S, F; nature of the bond (double band Dbl, triple band Tpl), number of cycles (Cyc) are considered for encoding



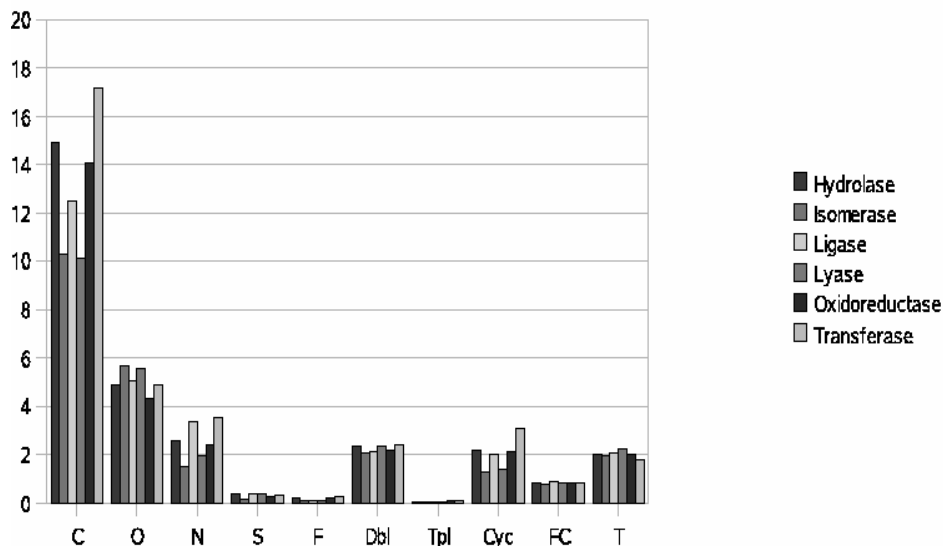


Fig. 2. Ligand features average values across the six enzyme classes is shown

the protein-ligand co-complex. These averages are shown in Figure 2. Except for count of C no other feature seems to be prominent in these features.

## 4 Classification and Discussion

The datasets comprise feature vectors grouped as six enzyme classes of proteins as given in Table 1. As can be seen, the numbers of proteins/binding pockets of the six classes are not uniform. Decision tree (J48) classifier, Random Forest (RF) and Support Vector Machine (SVM) are used to predict the protein class for known protein complexes. Here 10-fold cross-validation is used to obtain classification accuracies. Prediction accuracy for a class is obtained by averaging over all the ten folds to arrive at average accuracy for the class. Then weighted accuracy is computed over all the classes. The performance of SVM does not improve over Decision tree (DT) and random forest (RF) and hence only the results of DT and RF are discussed below.

Decision tree is a classical classifier which ranks the features according to information entropy and divides the dataset into six classes such that error of classification is minimized. For this experiment, J48 decision tree classifier is chosen which uses information gain to rank the features and splits the node at each level [Mitchell *et al.*, 1997]. We also train a Random Forest (RF) classifier containing ten decision trees to assess if the performance can be improved. RF is basically a composition of two or more decision trees and the decision is taken by consensus on the individual trees. The results of accuracy the average over all the ten folds for the six enzyme classes using and Random Forest (RF) and Decision Tree

**Table 2.** Random Forest, Here the set containing binding features (BS), fraction of contact (FC), tightness (T), ligand features (LF)

Protein Class	BS	BS + FC + T	LF + BS + FC + T	AAP	DPP
Hydrolases	0.75	0.76	0.79	<b>0.84</b>	0.82
Isomerases	0.13	0.11	0.17	0.39	0.39
Ligases	0.09	0.11	0.13	0.34	0.29
Lyases	0.24	0.23	0.22	0.43	0.38
Oxidoreductases	0.42	0.40	0.43	<b>0.64</b>	0.64
Transferases	0.65	0.63	0.63	<b>0.76</b>	0.76
Weighted Average	0.56	0.55	0.57	<b>0.70</b>	0.69

**Table 3.** Classification with Decision tree with features constituting binding features (BS), fraction of contact (FC), tightness (T), ligand features (LF)

Protein Class	BS	BS + FC + T	LF + BS + FC + T	AAP	DPP
Hydrolases	0.61	0.61	0.58	<b>0.74</b>	0.74
Isomerases	0.17	0.16	0.14	0.37	0.37
Ligases	0.14	0.12	0.15	0.33	0.27
Lyases	0.27	0.26	0.23	0.36	0.37
Oxidoreductases	0.35	0.36	0.35	0.53	0.58
Transferases	0.58	0.55	0.56	<b>0.71</b>	0.70
Weighted Average	0.48	0.48	0.46	<b>0.62</b>	0.63

(DT) are given in tables Table 2 and Table 3 respectively. All these experiments are carried out using Weka toolkit [Hall *et al.*, 2009]. We carried out the classification procedure by encoding the binding pocket using the additional features of fraction of contact and average tightness. We note that the additional features only add marginally to the percentage of accuracy. In order to compare with the earlier work in literature, we have conducted the entire set of experiments using the AA features derived from the whole protein (AAP) and didpeptide (bigram) features (DPP) and these are included in tables Table 2 and Table 3. An average weighted accuracy of approximately 48% is achieved with decision tree classifier and random forest improves the results by predicting with a weighted accuracy of 56%.

## 5 Conclusions

In this work, enzyme classification is carried out using specially features from the binding site as well as ligands and their mutual interaction. Since there exists an imbalance in the data available for different enzyme classes we compute a weighted accuracy of classification which is nearly 48% and 56% using classifiers decision tree (DT) and random forest (RF) respectively. Enzyme classification turns out to be a challenging problem if the information is only derived from

the structures. Clearly if the sequence features of the whole protein sequence are used, accuracy goes up to nearly 70%. The sparsity of data for some classes like ligases and lyases drops the prediction accuracy for those classes and the data imbalance amongst the classes compounds the issue further. The original hypothesis of how the structural features can be used to derive functional (EC) annotation of the enzyme needs to be investigated further.

**Acknowledgement.** GNS thanks DBT and DST for funding the project.

## References

- [Bork *et al.*, 1998] Bork, P., Koonin, E.V.: Predicting functions from protein sequences where are the bottlenecks? *Nat. Genet.* 18, 313–318 (1998)
- [Bray *et al.*, 2009] Bray, T., Doig, A.J., Warwicker, J.: Sequence and Structural Features of Enzymes and their Active Sites by EC Class. *J. Mol. Biol.* 386, 1423–1436 (2009)
- [Devos *et al.*, 2000] Devos, D., Valencia, A.: Practical Limits of Function Prediction. *PROTEINS: Str., Fun. and Genetics* 41, 98–107 (2000)
- [Dobson *et al.*, 2005] Dobson, P.D., Doig, A.J.: Predicting enzyme class from protein structure without alignments. *J. Mol. Biol.* 345, 187–199 (2005)
- [Malik *et al.*, 2007] Malik, A., Ahmad, S.: Sequence and structural features of carbohydrate binding in proteins and assessment of predictability using a neural network. *BMC Structural Biology* 7(1) (2007)
- [Mitchell *et al.*, 1997] Mitchell, T.M.: *Machine Learning*, vol. 52. McGraw-Hill Series in Comp. Sci., New York (1997)
- [PDB] Protein Data Bank, <http://www.pdb.org>
- [Porter *et al.*, 2004] Porter, C.T., Bartlett, G.J., Thornton, J.M.: The Catalytic Site Atlas: a resource of catalytic sites and residues identified in enzymes using structural data. *Nucleic Acids Res.* 32, D129–D133 (2004)
- [Reddy *et al.*, 2008] Reddy, A.S., Amarnath, H.S.D., Bapi, R.S., Sastry, G.M., Sastry, G.N.: Protein ligand interaction database (PLID). *Comp. Biol. and Chem.* 32, 387–390 (2008)
- [Rost *et al.*, 2003] Rost, B., Liu, J., Nair, R., Wrzeszczynski, K.O., Ofran, Y.: Automatic prediction of protein function. *Cell. Mol. Life Sci.* 60, 2637–2650 (2003)
- [Shen *et al.*, 2007] Shen, H.-B., Chou, K.-C.: EzyPred: A top down approach for predicting enzyme functional classes and subclasses. *Biochemical and Biophysical Research Communications* 364, 53–59 (2007)
- [Soga *et al.*, 2007] Soga, S., Shirai, H., Kobori, M., Hirayama, N.: Use of Amino Acid Composition to Predict Ligand-Binding Sites. *J. Chem. Inf. Model.* 47, 400–406 (2007)
- [Watson *et al.*, 2004] Watson, J.D., Sanderson, S., Ezersky, A., Savchenko, A., Edwards, O.C., Joachimiak, A., Laskowski, R.A., Thornton, J.M.: Towards fully automated structure-based function prediction in structural genomics: a case study. *J. Mol. Biol.* 367, 1511–1522 (2007)
- [Hall *et al.*, 2009] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA Data Mining Software: An Update. *SIGKDD Explorations* 11(1) (2009), <http://www.cs.waikato.ac.nz/ml/weka/>
- [Irwin *et al.*, 2005] Irwin, J.J., Shoichet, B.K.: ZINC - A Free Database of Commercially Available Compounds for Virtual Screening. *J. Chem. Inf. Model.* 45(1), 177–182 (2005), <http://www.zinc.docking.org>

# Non-linear Grayscale Image Enhancement Based on Firefly Algorithm

Tahereh Hassanzadeh, Hakimeh Vojodi, and Fariborz Mahmoudi

Faculty of IT and Computer Engineering Qazvin Azad University, Qazvin, Iran  
{t.hassanzadeh,h.vojodi,mahmoudi}@qiau.ac.ir

**Abstract.** The principal objective of enhancement is to improve the contrast and detail an image so, that the result is more suitable than the original image for a specific application. The enhancement process is a non-linear optimization problem with several constraints. In this paper, an adaptive local enhancement algorithm based on Firefly Algorithm (FA) is proposed. FA represents a new approach for optimization. The FA is used to search the optimal parameters for the best enhancement. In the proposed method, the evaluation criterion is defined by edge numbers, edge intensity and the entropy. The proposed method is demonstrated and compared with Linear Contrast Stretching (LCS), Histogram Equalization (HE), Genetic Algorithm based image Enhancement (GAIE), and the Particle Swarm Optimization based image enhancement (PSOIE) methods. Experimental results presented that proposed technique offers better performance.

**Keywords:** image enhancement, Firefly Algorithm, evaluation criterion, entropy.

## 1 Introduction

Image enhancement, one of the most important image processing techniques, can be treated as transforming one image to another to improve the interpretability of information for viewers, or to produce a processed image that is suitable for a given application [1]. For example, we might require an image that is easily recognized by a human observer or an image that can be analyzed and interpreted by a computer. There are two different strategies to achieve this goal. First, the image can be displayed appropriately so that the conveyed information is maximized. This will help a human (or computer) extract the desired information. Second, the image can be processed so that the informative part of the data is retained and the rest discarded [2]. Image enhancement techniques can be divided into four main categories: point domain, spatial domain, transformation, and pseudo coloring. The work done in this paper is based on spatial domain. Spatial domain refers to the image plane itself, and approaches in this category are based on direct manipulation of pixels in an image [1].

Histogram transformation is considered as one of the fundamental processes for contrast enhancement of gray level images [3], which facilitates subsequent higher level operations such as detection and identification. Recently, most digital color

image processing was done at the pseudo color level. Evolutionary algorithms have been previously used to perform image enhancement, for example, in [4] applied a global contrast enhancement technique using genetic programming (GP) to adapt the color map in the image so as to fit the demands of the human interpreter. A real coded GA is used with a subjective evaluation criterion to globally adapt the gray level intensity transformation in the image [5]. Combinations of different transformation functions with different parameters are used to produce the enhanced image by GA [6]. In [7], [14] performed gray level image contrast enhancement by PSO.

One of the modern heuristic algorithms that can be applied to non linear and non continuous optimization problems is the Firefly Algorithm (FA) [8], [13]. FA has some characteristics that make it suitable for solving optimization problem, like higher converging speed and less computation rate. In this paper, to improve the detail and gray scale of images, we have performed gray level image contrast enhancement by FA. Where, FA is used to find suitable parameters for image enhancement. Both objective and subjective evaluations are performed on the resulted image. As an objective criterion we use entropy and number of edges. The resulted gray level enhanced images by FA are compared with LCS, HE, GAIE and PSOIE. The experimental results and evaluations, shows that the efficiency and goodness of proposed enhancement method compared with mentioned automatic enhancement method.

## 2 Image Enhancement

Image enhancement done on spatial domain uses a transform function that is based on the gray level distribution. Local enhancement method applies transformation on a pixel considering intensity distribution among its neighboring pixels. The function used here is designed in such a way that takes both global as well as local information to produce the enhanced image. Local information is extracted from a user defined window of size  $n \times n$ . spatial domain processes will be denoted by the expression:

$$g(x, y) = T[f(x, y)] \tag{1}$$

where,  $f(x, y)$  is the input image,  $g(x, y)$  is the processed image, and  $T$  is an operator on  $f$ , defined over some neighborhood of  $(x, y)$  [1]. The transformation  $T$  is defined as:

$$g(i, j) = K(i, j)[f(i, j) - c \times m(i, j)] + m(i, j)^a \tag{2}$$

$a$ , and  $c$  are two parameters,  $m(i, j)$  is the local mean of the  $i, j$  pixel of the input image over a  $n \times n$  window and  $K(i, j)$  is enhancement function which takes both local and global information into account, Expression for local mean and enhancement function are defined as:

$$m(i, j) = \frac{1}{n \times n} \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} f(x, y) \tag{3}$$

One form of the enhancement function, used in this paper is:

$$K(i, j) = \frac{k.D}{\sigma(i, j) + b} \tag{4}$$

where,  $k$  and  $b$  are two parameters,  $D$  is the global mean and  $\sigma(i, j)$  is the local standard deviation of  $(i, j)$  pixel of the input image over a  $n \times n$  window, which are defined as:

$$D = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} f(i, j) \tag{5}$$

where,  $M$  and  $N$  are size of original image.

Thus the transformation function looks like:

$$\sigma(i, j) = \sqrt{\frac{1}{n \times n} \sum_{x=0}^n \sum_{y=0}^n (f(x, y) - m(i, j))^2} \tag{6}$$

$$g(i, j) = \frac{k.D}{\sigma(i, j) + b} [f(i, j) - c \times m(i, j)] + m(i, j)^a \tag{7}$$

Contrast of the image is stretched considering local mean as the center of stretch. Four parameters  $a$ ,  $b$ ,  $c$ , and  $k$  are introduced in the transformation function, to produce large variations in the processed image.

### 3 The Firefly Algorithm

The Firefly Algorithm (FA) developed by Xin-She Yang at Cambridge University [8]. In the FA, there are three idealized rules: 1) all fireflies are unisex so that one firefly will be attracted to other fireflies regardless of their sex. 2) Attractiveness is proportional to their brightness, thus for any two flashing fireflies, the less bright one will move towards the brighter one. 3) The brightness of a firefly is determined by the landscape of the objective function. As one of firefly attractiveness is should select any monotonically decreasing function. The distance intra any two fireflies  $i$  and  $j$  at  $x_i$  and  $x_j$ , respectively, is the Cartesian distance:

$$r_{ij} = \| x_i - x_j \| = \sqrt{\sum_{k=1}^d (x_{i,k} - x_{j,k})^2} \tag{8}$$

As a firefly’s attractiveness is proportional to the light intensity seen by adjacent fireflies, we can now define the attractiveness  $\beta$  of a Firefly by:

$$\beta(r) = \beta_0 e^{-\gamma r^2} \tag{9}$$

where, the  $\beta_0$  is the attractiveness at  $r = 0$  and  $\gamma$  is the light absorption coefficient at the source. It should be noted that the  $r$  is the Cartesian distance between any two fireflies  $i$  and  $j$  at  $x_i$  and  $x_j$ , where,  $x_i$  and  $x_j$  are spatial coordinate of fireflies  $i$  and  $j$ :

$$X_i = X_i + \beta_0 e^{-\gamma_{ij}} (X_j - X_i) + \alpha (\text{rand} - \frac{1}{2}) \quad (10)$$

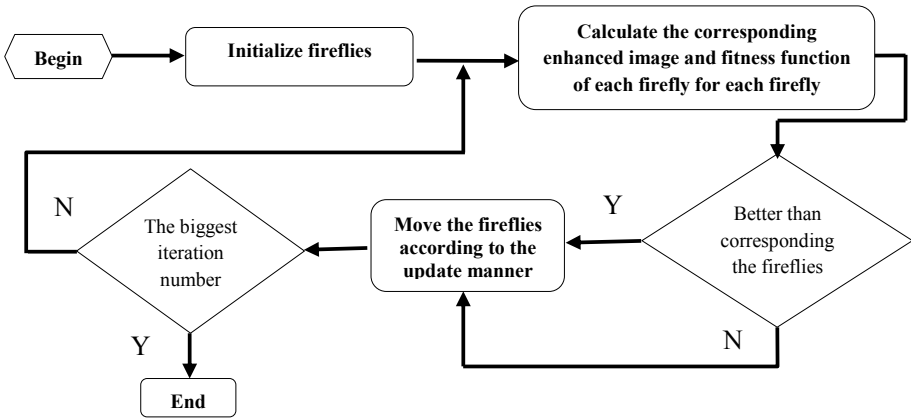
where, the second term is due to the attraction while the third term is a randomization with use of randomization parameter ( $\alpha$ ). In our implementation, we can take  $\beta_0=1$  and  $\alpha \in [0, 1]$ . Rand is a random number generator uniformly distributed in  $[0, 1]$ . The parameter  $\gamma$  characterizes the variation of the attractiveness, and its value is important in determining the speed of the convergence and how the FA behaves. In most applications, it typically varies from “0.01” to “100”. In this paper we set  $\gamma=1$ .

## 4 The Proposed Method

In the proposed enhancement method, to produce an enhanced image a transformation function defined in eq. (7) is used, which incorporates both global and local information of the input image. The function also have four parameters namely,  $a$ ,  $b$ ,  $c$ , and  $k$  which are used to produce diverse result and help to find the optimal image according to the objective function. The range of these parameters are,  $a \in [0, 1.5]$ ,  $b \in [0, 0.5]$ ,  $c \in [0, 1]$  and  $k \in [0.5, 1.5]$  according to [12]. In the proposed method, FA used to find the best values for mentioned parameters. Each firefly in the proposed technique is initialized with four parameters  $a$ ,  $b$ ,  $c$ , and  $k$ . firefly's values set randomly in the mentioned range. Our implementation, we can take  $\beta_0=1$  and  $\alpha \in [0, 1]$ . Rand is a random number generator uniformly distributed in  $[0, 1]$  and  $\lambda=1.5$ . Each firefly generates an enhanced image with defined parameters. Quality of the enhanced image is calculated by an objective function defined in eq. (8). The fitness of each firefly is corresponding to quality of enhanced image and the movement of each firefly is based on the fitness function. The flowchart of the proposed algorithm is shown in Fig. 1.

## 5 Experimental Results

The optimization problem considered in this paper is to solve the enhancement problem using FA. Our objective is to maximize the number of pixels in the edges, increase the overall intensity of the edges, and increase the measure of the entropy. In the proposed method, the FA parameters set according to [8]. Population size is set as 25. The fireflies are initialized as a value  $a$ ,  $b$ ,  $c$  and  $k$ , which discussed in previous section and the largest truncated generation, is 30. In order to demonstrate the performance of the FA based image enhancement, comparison with linear contrast stretching (LCS), histogram equalization (HE), GA based image enhancement (GAIE) and PSO based image enhancement (PSOIE) is conducted. We test our approach using three selected images. They are the “Duck”, “Lady” and “Hut”. The properties of the pictures and the result of optimization problem show in Table 1.



**Fig. 1.** Flowchart of proposed enhancement method

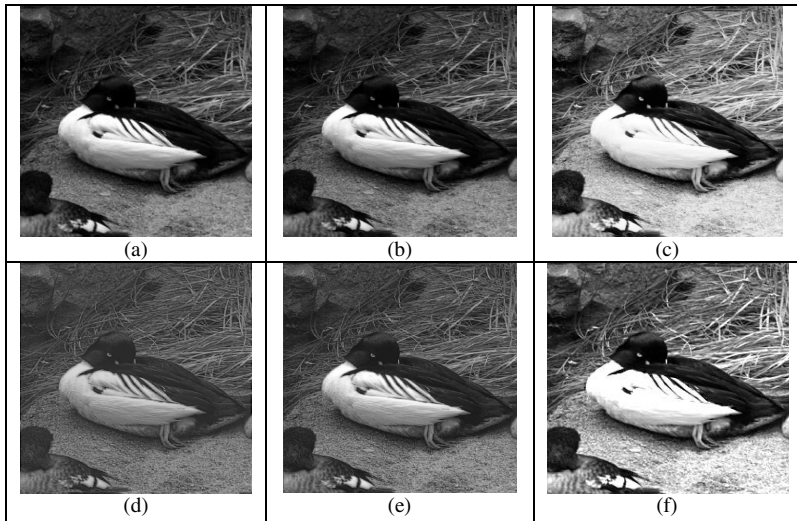
$P$ ,  $t$  and  $w$  in third column of Table 1 signify the number of particles, maximum number of generations and window size taken to extract the local information. Also, the optimal enhancement parameters, which found by the proposed method, have been shown in next four columns. It can be shown from Fig. 2, 3 and 4, that the brightness and contrast of proposed method results appear visibly and is more than the brightness and contrast of the original images. Also, it can be shown clearly, that the brightness of the enhanced images using FA is better than the brightness of the enhanced images using four other methods.

**Table 1.** The properties of original images

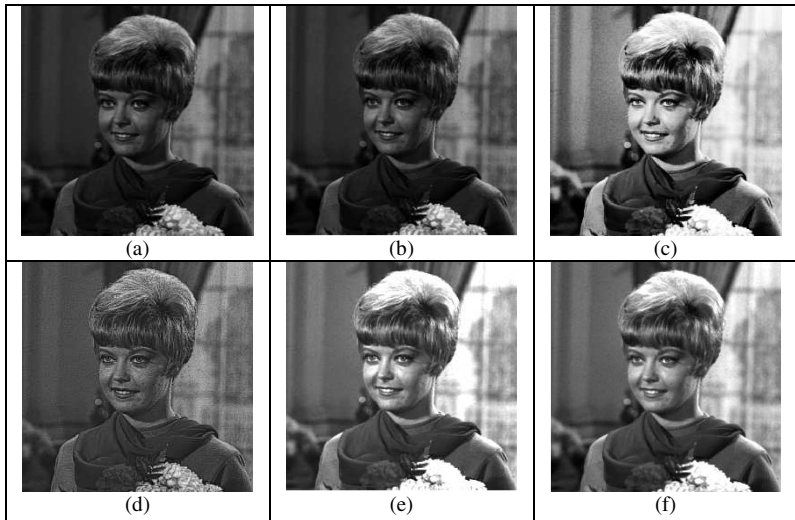
Images	Size (MxN)	P/t/w	a	b	c	k
Duck	256x256	25/30/5	0.8058	0.1488	0.7426	0.9988
Hut	256x256	25/30/5	1.1361	0.4451	0.9082	1.4563
Lady	256x256	25/30/5	0.8185	0.2826	0.3998	1.2274

If we visually analyze the images then we will see that, in the “Duck” image, Fig. 2, HE produced a good contrast result but it got so bright and loose some detail of image, GAIE produced a faded image and PSOIE result is also a little faded but as it shown, our proposed method result is so visible and clear. In “Lady” image, Fig. 3, HE and PSOIE are so bright and loose some detail of image, but the proposed enhanced image is bright and visible. Similarly in “Hut” image, Fig. 4, is clearly visible in FA based enhanced image. Also GAIE provided a clear view but the image became little blur. PSOIE result is also a little blur. Overall subjective evaluation of experimental results show the capability of the FA based image enhancement.





**Fig. 2.** Duck Image, (a) Original Image, (b) LCS, (c) HE, (d) GAIE, (e) PSOIE and (f) Proposed FA based Method



**Fig. 3.** Lady Image, (a) Original Image, (b) LCS, (c) HE, (d) GAIE, (e) PSOIE and (f) Proposed FA based Method

In order to objective evaluation we used number of edgels and entropy measure as a criterion. Increasing of these two criteria indicate the goodness and the performance of resulted images. As shown in Table 2 that the FA-based method achieves the best detail content in the enhanced images when compared with the number of edgels in the enhanced image using LCS, HE, GAIE and PSOIE. The edges detected with a

Sobel edge detector. This ensures that the FA method yields better quality of solution compared to mentioned methods. As it shown in Table 3 to evaluate the proposed method results we also used entropy measure. Entropy indicates the detail and randomness of image, as it mentioned in Table 3 we increase the entropy measure of images with proposed technique. All these evaluation extend the capability of the FA based gray scale image enhancement.

**Table 2.** The number of edgels as detected with sobel automatic edge detector for enhanced images

Images	Original	LCS	HE	GAIE	PSOIE	FA
Duck	1635	1607	1782	1455	1417	<b>2028</b>
Hut	1944	1992	2099	892	1820	<b>2367</b>
Lady	1286	1413	2187	2515	1810	<b>2823</b>



**Fig. 4.** Hut Image For all images: (a) Original Image, (b) LCS, (c) HE, (d) GAIE, (e) PSOIE and (f) Proposed FA based Method

**Table 3.** Entropy values of the enhanced images

Images	Original	LCS	HE	GAIE	PSOIE	FA
Duck	7.5308	7.5279	7.3846	6.9813	7.2817	<b>7.9635</b>
Hut	7.5592	7.5338	7.0220	6.9406	7.5154	<b>7.6953</b>
Lady	7.0697	7.2003	7.1908	7.0134	7.5392	<b>7.8016</b>

## 6 Conclusion

In this paper, we have proposed a FA based grayscale image enhancement technique. The objective of the algorithm was to maximize the total number of pixels in the edges, increase the overall intensity of the edges and entropy measures thus being able to visualize more details in the images. Results of the proposed technique are compared with some other image enhancement techniques, like linear contrast

stretching, histogram equalization and genetic algorithm based image enhancement and PSO based image enhancement. Both subjective and objective evaluations prove the capability of proposed algorithm to enhance the images.

## References

1. Gonzalez, R.C., Woods, R.E.: Digital Image Processing. Addison-Wesley, New York (1992)
2. Galatsanos, N.P., Segall, C.A., Katsaggelos, A.K.: Digital Image Enhancement. In: Encyclopedia of Optical Engineering, doi:10.1081/E-EOE 120009510
3. Gonzalez, C., Fittes, B.A.: Gray-level transformations for interactive image enhancement. *Mechanism and Machine Theory* 12, 111–122 (1977)
4. Bck, T., Fogel, D., Michalewicz, Z.: Handbook of Evolutionary Computation. Oxford Univ. Press, London (1997)
5. Munteanu, C., Lazarescu, V.: Evolutionary contrast stretching and detail enhancement of satellite images. In: Proc. Mendel, Berno, Czech Rep., pp. 94–99 (1999)
6. Pal, S.K., Bhandari, D.M., Kundu, K.: Genetic algorithms for optimal image enhancement. *Pattern Recognition Letter* 15, 261–271 (1994)
7. Gorai, A., Ghosh, A.: Gray-level Image Enhancement by Particle Swarm Optimization. In: World Congress on Nature & Biologically Inspired Computing, 978-1-4244-5612 (2009)
8. Braik, M., Sheta, A., Ayesh, A.: Image Enhancement Using Particle Swarm Optimization. In: WCE 2007, London, U.K. (2007)
9. Xiang, Z., Yan, Z.: Algorithm based on local variance to enhance contrast of fog-degraded image. *Computer Applications* 27, 510–512 (2007)
10. Munteanu, C., Rosa, A.: Gray-scale enhancement as an automatic process driven by evolution. *IEEE Transaction on Systems, Man and Cybernetics-Part B: Cybernetics* 34(2), 1292–1298 (2004)
11. Yang, X.-S.: Firefly algorithm, stochastic TestFunctions and Design Optimization. *Int. J. Bio-Inspired Computation* 2(2), 78–84 (2010)
12. Venkatalakshmi, K., Mercy Shalinie, S.: A Customized Particle Swarm Optimization Algorithm for Image Enhancement. In: ICCCT 2010, 978-1-4244-7770 (2010)
13. Yan, X.S.: Nature-Inspired Metaheuristic Algorithms. LuniverPress (2008)
14. Munteanu, C., Rosa, A.: Gray-scale enhancement as an automatic process driven by evolution. *IEEE Transaction on Systems, Man and Cybernetics-Part B: Cybernetics* 34(2), 1292–1298 (2004)

# Synthesis and Design of Thinned Planar Concentric Circular Antenna Array - A Multi-objective Approach

Sk. Minhazul Islam<sup>1</sup>, Saurav Ghosh<sup>1</sup>, Subhrajit Roy<sup>1</sup>, Shizheng Zhao<sup>2</sup>,  
Ponnuthurai Nagaratnam Suganthan<sup>2</sup>, and Swagamtam Das<sup>1</sup>

<sup>1</sup>Dept. of Electronics and Telecommunication Engg. ,  
Jadavpur University, Kolkata 700 032, India

<sup>2</sup>Dept. Of Electronics and Electrical Engg.,  
Nanyang Technological Univrsity  
{skminha.isl,roy.subhrajit20}@gmail.com,  
saurav\_online@yahoo.in, ZH0047NG@e.ntu.edu.sg,  
epnsugan@ntu.edu.sg, swagatamdas19@yahoo.co.in

**Abstract.** Thinned concentric antenna array design is one of the most important electromagnetic optimization problems of current interest. This antenna must generate a pencil beam pattern in the vertical plane along with minimized side lobe level (SLL) and desired HPBW, FNBW and number of switched off elements. In this article, for the first time to the best of our knowledge, a multi-objective optimization framework for this design is presented. Four objectives described above we are treated as four distinct objectives that are to be optimized simultaneously. The multi-objective approach provides greater flexibility by yielding a set of equivalent final solutions from which the user can choose one that attains a suitable trade-off margin as per requirements. In this article, we have used a multi-objective algorithm of current interest namely the *NSGA-II* algorithm. There are two types of design, one with uniform inter-element spacing fixed at  $0.5\lambda$  and the other with optimum uniform inter-element spacing. Extensive simulation and results are given with respect to the obtained HPBW, SLL, FNBW and number of switched off elements and compared with two state-of-the-art single objective optimization methods namely DE and PSO.

## 1 Introduction

Circular antenna array, in which antenna elements are placed in a circular ring, is an array configuration of very practical use among all other antenna arrays present in modern day. It consists of a number of elements arranged on a circle [1] with uniform or non-uniform spacing between them. It possesses various applications in sonar, radar, mobile and commercial satellite communications systems [1-3]. Concentric Circular Antenna Array (CCAA), one of the most important circular arrays, contains many concentric circular rings of different radii and number of elements proportional to the ring radii. Uniform CCA (UCCA) is one of the most important configurations

of the CCA [2] where the inter-element spacing in individual ring is kept almost half of the wavelength and all the elements in the array are uniformly excited.

In Concentric circular array, for reduction of the side lobe level, the array must be made aperiodic by altering the positions of the antenna elements. Thinning a large array will not only reduce side lobe level further but also reduce the number of antennas in the array and thereby cut down cost substantially. Global optimization tools such as Genetic Algorithms (GA) [6], Particle Swarm Optimization (PSO) [7], and Differential Evolution (DE) [9, 10] etc. have been used to solve these problems. In this article, for the first time to the best of our knowledge we have proposed a multi-objective framework [4] for the design of thinned concentric circular antenna array. Instead of going for the single objective weighted sum method which is however, subjective and the solution obtained will depend on the values of the weights specified, motivated by the inherent multi-objective nature of the antenna array design problems and the overwhelming growth in the field of Multi-Objective Evolutionary Algorithms [14], we have opted for a multi-objective algorithm named NSGA-II [11] to solve the thinned concentric circular array synthesis problem much more efficiently as compared to the conventional single-objective approaches like many other MO approaches previously [12-13]. This MO framework attempts to achieve a suitable number of switched off element in the thinned array close to the desired value, a desired half power beamwidth (HPBW) and first null beamwidth (FNBW) and also the a reduced side-lobe level (SLL). In order to show the effectiveness *NSGA-II* algorithm we have compared the obtained results of the *NSGA-II* algorithm with two traditional single objective evolutionary algorithms, DE [9,10] and PSO [7,8] which is outperformed by the NSGA-II in terms of the SLL, HPBW, FNBW and number off switched off elements.

## 2 General Description of NSGA-II

NSGA-II [11] is a non-domination based genetic algorithm for multi-objective optimization which incorporates elitism and no sharing parameter needs to be chosen *a priori*. The population is initialized as usual. Once the population is initialized the population is sorted based on non-domination into each front. The first front being completely non-dominant set in the current population and the second front being dominated by the individuals in the first front only and the front goes so on. Each individual in the each front are assigned rank (fitness) values or based on front in which they belong to. Individuals in first front are given a fitness value of 1 and individuals in second are assigned fitness value as 2 and so on. In addition to fitness value a new parameter called *crowding distance* is calculated for each individual. The crowding distance is a measure of how close an individual is to its neighbours. Large average crowding distance will result in better diversity in the population. Parents are selected from the population by using binary tournament selection based on the rank and crowding distance. An individual is selected in the rank is lesser than the other or if crowding distance is greater than the other. The selected population generates off-springs from crossover and mutation operators. The population with the current

population and current off-springs is sorted again based on non-domination and only the best  $N$  individuals are selected, where  $N$  is the population size. The selection is based on rank and the on crowding distance on the last front.

### 3 Design of Thinned Concentric Circular Array: The Proposed Multi-objective Framework

Thinning an array means turning off some elements in a uniformly spaced or periodic array to generate a pattern with low side lobe level. In our method, we kept the antennas positions fixed, and all the elements can have only two states either “on” or “off” (Similar to Logic “1” and “0” in digital domain). An antenna will be considered to be in “on” state if and only if it contributes to the total array pattern. While an antenna will be considered “off” if and only if either the element is passively terminated to a matched load or open circuited. If an antenna element does not contribute to the resultant array pattern, they will be considered “off”. As for non-uniform spacing of the element one has to check an infinite number of possibilities before final placement of the elements, thinning an array [12-14] to produce low side lobes is much simpler than the more general problem of non-uniform spacing of the elements. The arrangement of elements in planar circular arrays [2, 3] may contain multiple concentric circular rings, which differ in radius and number of elements. Figure 1 shows the configuration of multiple concentric circular arrays [2, 3] in  $XY$  plane in which there are  $M$  concentric circular rings. The  $m$ -th ring has a radius  $r_m$  and number of isotropic elements  $N_m$  where  $m = 1,2,3, \dots, 10$ . Elements are equally placed along a common circle. The far-field pattern [1] in free space is given by:

$$E(\theta, \phi) = \sum_{m=1}^M \sum_{n=1}^{N_m} I_{mn} e^{j2\pi r_m \sin \theta \cos(\phi - \phi_{mn})} \tag{1}$$

Normalized power pattern in dB denoted by  $P(\theta, \phi)$  can be expressed as follows:

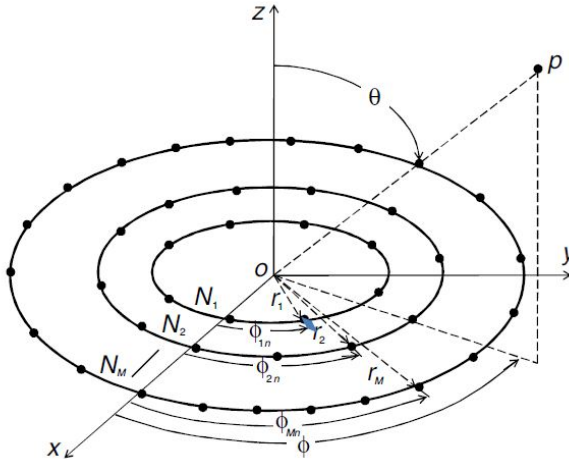
$$P(\theta, \phi) = 10 \log \left[ \frac{|E(\theta, \phi)|}{|E(\theta, \phi)_{\max}|} \right]^2 = 20 \log \left[ \frac{|E(\theta, \phi)|}{|E(\theta, \phi)_{\max}|} \right] \tag{2}$$

$r_m$  = radius of the  $m$ -th ring =  $\frac{N_m d_m}{2\pi}$ ,  $d_m$ =inter-element distance of the  $m$ -th ring.

$\phi_{m,n} = \frac{2n\pi}{N_m}$  = angular position of the  $mn$ -th element with

$1 < n \leq N_m$ ,  $\theta, \phi$  = polar and azimuthal position of  $mn$ -th element with  $= \frac{2\pi}{\lambda}$ ,  $\lambda$  = wave-length.  $I_{m,n}$  = excitation amplitude of  $mn$ -th element. In our case

the excitation amplitude of the an element is said to be turned “on” when it takes a value of “1” and is said to be turned “off” if it takes a value of “0”. All the elements have the same excitation phase of zero degree. The value of  $\phi$  is taken as  $0^0$ .



**Fig. 1.** Multiple concentric circular ring arrays of isotropic antennas in XY plane

There are four objectives for the multi-objective framework. They are the reduced maximum SLL, desired HPBW (-4.5dB), desired FNBW (15 degree) and the desired no. of switched off elements. So, the following objectives are:

$$f_1 = (SLL_{max}), \quad f_2 = (HPBW_o - HPBW_r)^2, \quad (3)$$

$$f_3 = (FNBW_o - FNBW_r)^2 \text{ and } f_4 = (T_o^{off} - T_r^{off})^2 H(T)$$

where,  $SLL_{max}$  is the value of maximum side lobe level.  $HPBW_o$ ,  $HPBW_r$  are obtained and desired value of half-power beam width respectively.  $FNBW_o$ ,  $FNBW_r$  are obtained and desired value of first null beam width respectively.  $T_o^{off}$ ,  $T_r^{off}$  are obtained and desired value of number of switched off element respectively.  $H(T)$  is Heaviside step functions defined as follows:

$$H(T) = \begin{cases} 0 & \text{if } T \leq 0 \\ 1 & \text{if } T > 0 \end{cases} \quad (4)$$

where  $T = T_o^{off} - T_r^{off}$ .

In this way, an MOEA will allow us to find the right balance between the four objectives shown above. So an MOEA will allow us greater flexibility in designing a thinned concentric circular antenna array because a single-objective EA gives us only one solution in one performance which might not completely satisfy the designer's needs.

## 4 Simulation and Results

The simulation and results for the design of the thinned concentric circular antenna array has been given on one instantiation of the design problem, namely ten concentric circular rings. In the example, each ring of the antenna contain  $8m$  equi-spaced isotropic elements (a total of 440), where  $m$  is the ring number counted from the innermost ring 1. The optimal inter-element arc spacing ( $d_m$ ), the Side-lobe level (SLL), the first null beamwidth (FNBW), the half power beam width (HPBW) and the optimal set of "on" and "off" are determined with respect to the best compromised solution which will be (HPBW) discussed below. For *NSGA-II*, the best compromise solution was chosen from the PF using the method described in [15]. Over the thinned circular concentric antenna array design instances and cases we also compare the performance of *NSGA-II* with that of two single-objective optimization techniques, namely the original DE [9] (DE/rand/Bin/1) and PSO [7] where objective function is the weighted sum of the four objectives (weights are taken as unity). Parameters for all the algorithms are selected with guidelines from their respective literatures. In what follows, we report the best results obtained from a set of 25 independent runs of *NSGA-II* and its competitors, where each run for each algorithm is continued up to  $3 \times 10^5$  Function Evaluations (FEs).

**Cases-I and II:** In this case, inter-element arc spacing ( $d_m$ ) in all the rings is  $0.5\lambda$ . For such a fully populated and uniformly excited array, the maximum side lobe level is calculated to be -17.37 dB and HPBW and the FNBW is taken as 4.5 and 15 degree. Problem is now to find the optimal set of "on" and "off" elements keeping the half-power beam width (HPBW) and the first null beamwidth (FNBW) unchanged and fixing the number of switched off elements to be equal to 220 or more and reducing the maximum side lobe level (SLL) further. Number of vectors is taken to be 100 and the algorithm is run for 50 generations. In the second case, inter-element arc spacing ( $d_m$ ) in all the rings is made uniform and same but not fixed. Optimum values of inter-element arc spacing along with optimal set of "on" and "off" elements are found out using this *NSGA-II* that will generate a pencil beam in the XZ plane with reduced side lobe level. Results in the Table 1 clearly show that the synthesized pattern of thinned array using *NSGA-II* and optimum inter-element arc spacing is better than a fully populated array in terms of side lobe level. Optimized inter-element arc spacing is found to be  $d = 0.4316\lambda$ . Table 2 depicts the excitation amplitude distributions for the two cases. Table 3 clearly shows that *NSGA-II* has outperformed



its competitor algorithms like DE and PSO. Here, figure 2 shows the normalized power pattern in dB scale for fully populated, optimized  $d = 0.4316\lambda$  and  $d = 0.5\lambda$ .

**Table 1.** Obtained results for Case I and Case II for *NSGA-II*

Design Parameters	Synthesized Thin Array with Optimum $d = 0.4316\lambda$	Synthesized Thin Array with fixed $d = 0.5\lambda$	Fully Populated Array with $d = 0.5\lambda$
Side Lobe level (SLL, in db)	<b>-21.29</b>	-19.61	-17.37
Half Power Beamwidth (HPBW, in degree)	<b>4.5</b>	<b>4.5</b>	<b>4.5</b>
First Null Beamwidth (FNBW, in degree)	<b>15</b>	<b>15</b>	<b>15</b>
Number of Switched Off Elements	<b>220</b>	<b>220</b>	0

**Table 2.** Excitation Amplitude Distributions ( $I_{mn}$ ) using *NSGA-II* with fixed  $d = 0.5\lambda$  and optimal  $d = 0.4316\lambda$  respectively

11000101
0101001110110011
101011001111010110111000
11010010100100011011101011010000
000001101010010010000000111000010100010
000011000100000011001001010011101010101101010001
0000000011011001100010000010010000111100001110101101111
1101000011011001111001011101111000011110111011000000011001001100
010111000110110110001100111101101010101010100101001111110110110011111000
0101000101110110111111011011110001110110110111010100110110001111001101001110
11001100
0010001111101010
111000110111011000111111
01010001001110110111100010101111
0010011001101000000100001001001011010100
000001000001100011110001100010100010111111111100
0001010000000101111101101100101101111101110001010010000
000101101101001111010100001100000011101111011001111000110011001
011111111100001100101100010010011101011010100100101100100010010100011001
01111000101110001010000101111100000100111110010010100110000100001111011010111110

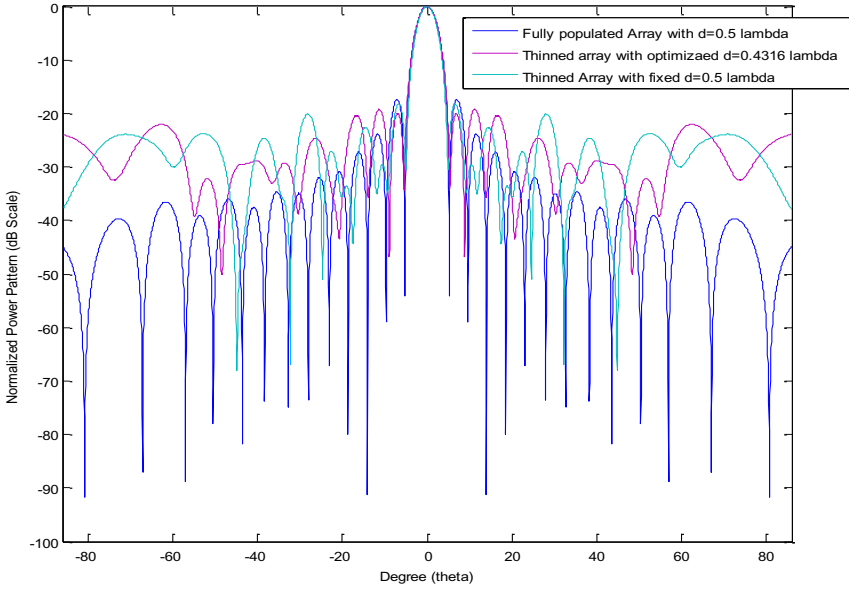


Fig. 2. Normalized power pattern in dB

Table 3. Set of comparison table for Case I and Case II

Design parameters	Synthesized thinned array with optimum value of $d = 0.5\lambda$			Synthesized thinned array with optimum value of $d$ obtained		
	Simple DE	PSO	NSGA-II	Simple DE	PSO	NSGA-II
Optimum Value of $d$	-	-	-	.5249	.4987	.4316
Side Lobe level (SLL, in db)	-15.32	-16.83	<b>-19.61</b>	-20.34	-21.03	<b>-21.29</b>
Half Power Beamwidth (HPBW, in degree)	3.8	4.0	<b>4.5</b>	3.5	3.7	<b>4.5</b>
First Null Beamwidth (FNBW, in degree)	13.45	14.21	<b>15</b>	12.69	13.98	<b>15</b>
Number of Switched Off Elements	198	206	<b>220</b>	195	205	<b>220</b>



12. Pal, S., Das, S., Basak, A., Suganthan, P.N.: Synthesis of difference patterns for monopulse antennas with optimal combination of array-size and number of subarrays - A multi-objective optimization approach. *Progress in Electromagnetics Research, PIER B* 21, 257–280 (2010)
13. Pal, S., Qu, B.Y., Das, S., Suganthan, P.N.: Optimal Synthesis of Linear Antenna Arrays with Multi-objective Differential Evolution. *Progress in Electromagnetics Research, PIER B* 21, 87–111 (2010)
14. Zhou, A., Qu, B.-Y., Li, H., Zhao, S.-Z., Suganthan, P.N., Zhang, Q.: Multi-objective Evolutionary Algorithms: A Survey of the State-of-the-art. *Swarm and Evolutionary Computation* 1(1), 32–49 (2011)
15. Abido, M.A.: A novel multi-objective evolutionary algorithm for environmental/economic power dispatch. *Electric Power Systems Research* 65, 71–81 (2003)

# Soft Computing Based Optimum Parameter Design of PID Controller in Rotor Speed Control of Wind Turbines

R. Manikandan\* and Nilanjan Saha\*\*

Department of Ocean Engineering,  
Indian Institute of Technology Madras, Chennai- 600036, India  
mkaucbe@gmail.com, nilanjan@iitm.ac.in

**Abstract.** Sensitivity and robustness is the primary issue while designing the controller for large non-linear systems such as offshore wind turbines. The main goal of this study is a novel soft computing based approach in controlling the rotor speed of wind turbine. The performance objectives for controller design is to keep the error between the controlled output (speed of rotor) and the target rotor speed, as small as possible. The wind turbine involves controlling both the aerodynamics and hydrodynamics response together, therefore in this paper an attempt is being made using soft computing approach. The commonly used proportional – integral – derivative controller (PID controller) for wind turbines employs Ziegler and Nichols (*ZN*) approach which leads to excessive amplitude in some situations. In this work, the parameters of PID controller are obtained using the conventional method that is *ZN* along with the artificial intelligence (AI) technique. Two types of AI (i) bacteria foraging optimization algorithm (BFOA) and (ii) particle swarm optimization (PSO) coupled with *ZN* controller are studied. The controller performance indices are taken as integral square error, steady state error, controller gain, maximum overshoot and settling time. In this work, the idea of model generation and optimization is explored for PID controller. The planned controller strategy would be able to carry out high quality performance which reveal that the proposed controller system can significantly reduce the errors and settling time.

**Keywords:** Wind turbines, optimization, proportional–integral–derivative controller, bacteria foraging optimization algorithm, particle swarm optimization.

## 1 Introduction

Globally, wind power has seen the highest growth rate as a renewable generation capacity during recent years. Wind turbines can be placed either on land or offshore. The wind energy resource is a random resource and the controlled

---

\* Corresponding author.

\*\* Assistant Professor.

energy extraction from wind is being actively pursued. The wind power output for a 5 MW. wind turbine increases approximately linearly till mean wind speeds of  $12 \text{ ms}^{-1}$  with speeds more than  $25 \text{ ms}^{-1}$  as a situation for idling of turbine when no power is extracted. The mean wind speeds in range of  $12 - 16 \text{ ms}^{-1}$  i.e., around the rated wind speeds is recommendable for generation of electricity. Since the wind speed is a highly stochastic process therefore, the wind power demand changes and therefore the power is not controllable. In other words, the wind speed can be very high resulting in power generation that exceeds the demand of the load [1]. This might lead to the turbine exceeding its rotational speed rating and subsequent damage to the turbine. On the other hand, the wind speed can be too low for any power production and therefore alternative energy sources should be used [2]. The fact that one has no control over the energy source input, the unpredictability of wind and the varying power demand are more than enough concerns to justify the need for a controller, which will regulate all the parameters that need to be controlled for a matched operation of the wind turbine.

Nowadays, the new computing schemes have arrived, which mimic the nature of human being, called as artificial intelligence. Herein, the word intelligence [4,3] exploits lot of meaning such as ability to acquire, understand the problem and apply the knowledge. It embodies all the knowledge both conscious and unconscious, which are acquired through study and experience, thought, imagination, express and feeling emotions. The computing schemes are fundamentally different from numerical methods like trial and error techniques. It plays important role to solve complex and non linear problems and accuracy. Presently, AI based computing scheme is proposed because it may be difficult to optimize the gain parameters of the PID controllers via conventional methods like  $ZN$  and open loop methods. This work mainly concentrates on the efficient working of swarm intelligence like BFOA [5] and PSO [6,7] to design a PID controller for a wind turbine, to control the turbine rotor speed.

## 2 Wind Turbine

The kinetic energy,  $U$ , of a parcel of air of mass  $m$  flowing at upstream speed  $u_{up}$  in the axial direction (x-direction) of the wind turbine is given by

$$U = \frac{1}{2} m u_{up}^2 = \frac{1}{2} (\rho A x) u_{up}^2. \quad (1)$$

Here,  $A$  is the cross-sectional (swept) area of the wind turbine,  $\rho$  is the air density, and  $x$  is the thickness of the wind parcel. The power in the wind  $P_w$  is the time derivative of the kinetic energy and is given in (2), which represents the total power available for extraction, i.e.,

$$P_w = \frac{dU}{dT} = \frac{1}{2} \rho A u_{up}^2 \frac{dX}{dT} = \frac{1}{2} \rho A u_{up}^3. \quad (2)$$

The extracted power is usually expressed in terms of the wind turbine swept area  $A$ , because the upstream cross-sectional area is not physically measurable

as the cross-sectional area of the wind turbine. The fraction of actual power extracted to the theoretical available power in the wind by practical turbines is expressed by the coefficient of performance  $C_p$ . The actual mechanical power ( $P_m$ ) extracted can be written as:

$$P_m = C_p P_w = C_p \left( \frac{1}{2} \rho A u_{up}^3 \right) \quad (3)$$

The value of  $C_p$  is highly non-linear and varies with the wind speed, the rotational speed of the turbine, and the turbine blade parameters such as pitch angle. The control sequence maintains a constant angular speed and constant power  $P_m$ . Only the angular speed is given a feedback to accommodate the wind speed fluctuations because controlling the angular speed would control the aerodynamic torque ( $T_A$ ). The tip speed ratio  $\lambda$  is defined as the ratio between the rectilinear speed of the turbine tip,  $\omega R$ , and the wind speed  $u_{up}$  as

$$\lambda = \frac{\omega R}{u_{up}}. \quad (4)$$

Another variable used to evaluate the wind turbine performance is the coefficient of torque  $C_t$ . The torque coefficient  $C_t$ , is related to the power coefficient  $C_p$  through the parameter  $\lambda$ . The aerodynamic actual torque  $T_A$  developed by the rotor that turns the rotor shaft is related to the torque coefficient  $C_t$  by

$$T_A = \frac{1}{2} \rho A R C_t u_{up}^2. \quad (5)$$

If the torque coefficient is tuned, then the power produced by the turbine would also be tuned. The wind turbine mechanical power  $P_m$  is equal to the product of the aerodynamic torque  $T_A$  and the rotational speed, which is [9,10,11]

$$P_m = T_A \times \omega_m. \quad (6)$$

Using the above (4), (5) and (6), the rotational speed is controlled as described in the subsequent section through mathematical models. For further information, the reader is referred to [8,9,11].

### 3 System Description

In deriving a wind turbine mathematical model a specific constant-speed, variable pitch-control wind turbine was selected. One of the key factors is to find values for the constant parameters in the transfer functions representing the wind turbine system at operating conditions. The variation of the coefficient of torque  $C_t$ , with the pitch angle  $\beta$ , and the tip speed ratio,  $\lambda$ , is highly nonlinear and unique for each wind turbine [11]. The geometry and aerodynamic characteristics of the simulated wind turbine resemble those of a Grumman Windstream-33, 10 m diameter, 20 kW turbine [9].

### 3.1 Wind Turbine Mathematical Model

The wind turbine plant model was divided into two main parts. The first part was the wind turbine, which included a turbine rotor on a low-speed shaft a gearbox and high-speed shaft [9]. The second part was the electric generator. Figure 1 illustrates the general block diagram of the wind turbine system.

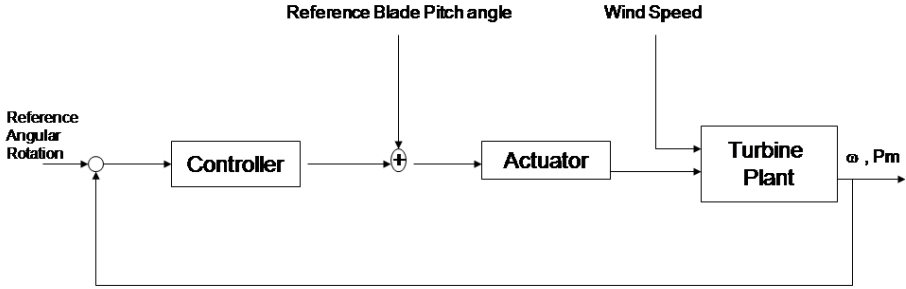


Fig. 1. Block Diagram of the Wind Turbine System

The goal of the control sequence is to maintain a constant angular speed and constant power  $P_m$ . Only the angular speed is fed back to accommodate the wind speed fluctuations because, controlling the angular speed would control the aerodynamic torque ( $T_A$ ). If the  $T_A$  that causes the rotation is controlled, the extracted mechanical power  $P_m$  is also done. This is derived from the fact that these three quantities,  $P_m$ ,  $T_A$  and  $\omega$  which is referred as [11]

$$\omega = \alpha \Delta u + \gamma \Delta \omega + \delta \Delta \beta \tag{7}$$

In this equation,  $\Delta \omega$ ,  $\Delta u$  and  $\Delta \beta$  represent deviations from the chosen operating point  $\omega_{op}$ ,  $u_{op}$  and  $\beta_{op}$ . The parameters  $\alpha$ ,  $\gamma$  and  $\delta$  represent the coefficients that represent the wind turbine dynamics at the linearization point. Applying Laplace transforms and simplifying (7) yields [11]:

$$\Delta \omega = [\alpha \Delta u(s) + \delta \Delta \beta(s)] \frac{1}{\delta - \gamma} \tag{8}$$

## 4 The Problem Formulation and Controller Design Methodology

Gain-selection for PID controllers have generally been a trial and error process relying on experience and intuition of control engineers. A systematic approach to gain-selection provides visualization of the potential performance enhancements to the system control [11]. This work presents a methodology for selecting gain values for a PID controller that regulates the rotor speed of constant-power wind turbine by adjusting the blade-pitch angle [14]. Performance of PID



depends on the gain parameters which is required to be optimized. Two different methods are used to tune the controller: conventional and AI based tuning method. The conventional open loop *ZN* method is used here as for the closed loop conventional method the undamped oscillations can cause excessive oscillations. The closed loop method also requires longer time to obtain a state of stable sustained response for a slow process.

#### 4.1 The Tuning of PID Controller Using Conventional Approach

**The conventional approach - Ziegler-Nichols Method.** The performance of a controller depends on the proper choice of tuning parameters. For a PID algorithm, the controller tuning would require selection of  $K_c$ ,  $T_i$  and  $T_D$  parameter values [12]. So it becomes necessary to tune the controller parameters to achieve desired performance with the proper choice of tuning constants. Without mention, such a choice would often be a subjective procedure and certainly process dependent. Among the existing methods, *ZN* is widely accepted method for tuning the PID controller [14]. Due to above reasons, it is planned to utilize the soft computing based approach because it is a artificial intelligence tool which differs from ‘hard’ computing in a sense that it is tolerant of imprecision, uncertainty, partial truth and approximation. BFOA and PSO are two soft computing methods used herein. The results are then compared to conventional techniques.

#### 4.2 The Tuning of PID Controller Using Soft Computing Based Approach

**Implementation of BFOA based PID controller.** BFOA can be applied to the tuning of PID controller gains to ensure optimal control performance at nominal operating conditions. Table 1 presents the BFOA parameter values and chosen for the tuning purpose are shown below. After giving the below parameters to BFOA the PID controllers can be easily tuned and thus system performance can be improved.

##### Algorithm of BFOA

Step 1. Initialize parameters:  $n, N, N_C, N_S, N_{re}, N_{ed}, P_{ed}, C(i)(i = 1, 2, \dots, N), \varphi^i$ , where

- $n$  : Dimension of the search space,
- $N$  : Number of bacteria in the population,
- $N_C$  : Chemotactic steps,
- $N_{re}$  : Number of reproduction steps,
- $N_{ed}$  : Number of elimination-dispersal events,
- $P_{ed}$  : Elimination-dispersal with probability,
- $C(i)$  : Size of the step taken in the random direction specified by the tumble.

Step 2. Elimination-dispersal loop :  $i = i + 1$

Step 3. Reproduction loop :  $k = k + 1$

Step 4. Chemotaxis loop :  $j = j + 1$

Substep A. For  $i = 1, 2, \dots, N$  take a chemotactic step for bacterium has follows

Substep B. Compute fitness function,  $ISE(i, j, k, l)$ .

Substep C. Let  $ISE_{last} = ISE(i, j, k, l)$  to save this value since we may find a better cost via a run.

Substep D. Tumble: generate a random vector  $\Delta(i) \in \mathbb{R}^n$  with each element  $1, 2, \dots, \Delta_m(i), m = 1, 2, \dots, p$ ,  $p$  a random number on  $[-1, 1]$ .

Substep E. Move: Let

$$\varphi^x(i + 1, j, k) = \varphi^x(i, j, k) + C(i) + \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \tag{9}$$

this results in a step of size  $C(i)$  in the direction of the tumble for bacterium  $i$ .

Substep F. Compute  $ISE(i, j + 1, k, l)$ .

Substep G. Swim

1. Let  $m = 0$  (counter for swim length).
2. While  $m < N_S$  (if have not climbed down too long).  
 Let  $m = m + 1$   
 If  $ISE(i, j + 1, k, l) < ISE_{last}$  (if doing better),  
 Let  $ISE_{last} = ISE(i, j + 1, k, l)$  and let

$$\varphi^x(i, j + 1, k) = \varphi^x(i, j, k) + C(i) + \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \tag{10}$$

and use this  $x(i + 1, j, k)$  to compute the new  $ISE(i, j + 1, k, l)$  as in [Sub-step F]. Else, let  $m = N_S$  this is the end of the while statement.

- Substep H. Go to next bacterium  $(i, j)$  if  $i \neq N$  (i.e., go to [Sub-step B] to process the next bacterium ).  
 Step 6. If  $j < N_C$ , go to Step 3. In this case continue chemotaxis, since the life of the bacteria is not over.  
 Step 7. Reproduction.  
 Substep A. : For the given  $k$  and  $l$ , and for each  $i = 1, 2, \dots, N$ , let

$$ISE_{health} = \sum_{j=1}^{N_C+1} ISE(i, j, k, l)$$

be the health of the bacterium  $i$  (a measure of how many nutrients it got over its lifetime and how successful it was at avoiding noxious substances). Sort bacteria and chemotactic parameters  $C(i)$  in order of ascending cost  $ISE_{health}$  (higher cost means lower health).

- Substep B; The  $S_r$  bacteria with the highest  $ISE_{health}$  values die and the remaining  $S_r$  bacteria with the best values split (this process is performed by the copies that are made placed at the same location as their parent)  
 Step 8. If  $k < N_{re}$ , go to [Step 3]. In this case, we have not reached the number of specified reproduction steps, so we start the next generation of the chemotaxis loop.

**Table 1.** Input details of BFOA Algorithm

BFOA Parameter	Value/Method
Dimension of search	2
Number of bacteria	10
Chematactic steps	5
Length of swim	4
Performance index / fitness function	ISE

**Implementation of PSO based PID controller.** Keeping the values of the PSO algorithm parameters inertia weight factor  $W = 0.3$  and the acceleration constants  $C_1 = C_2 = 1.5$  as fixed, the three gain parameters  $K_P, K_D, K_I$  of the controller are optimized. The search for the optimal value is in a three dimensional search space.

*Algorithm of BFOA*

- Step 1. The  $i^{th}$  particle in the swarm is represented as  $K_i = (K_{i1}, K_{i2}, K_{i3}, \dots, K_{id})$  in the d-dimensional space.

- Step 2. The best previous of the  $i^{th}$  particle is represented as

$$P_{best} = (P_{best_{i,1}}, P_{best_{i,2}}, P_{best_{i,3}}, \dots, P_{best_{i,d}})$$

where,

$d$  : Dimension index

$P_{best_i}$  : Best previous position of the  $i^{th}$  particle.

- Step 3. The index of the best particle among the group is  $G_{best}$  where,  
 $G_{best}$  : Best particle among all the particle in the swarming population.

- Step 4. Velocity of the  $i^{th}$  particle is represented as  $V_i = (V_{i,1}, V_{i,2}, V_{i,3}, \dots, V_{i,d})$

- Step 5. The updated velocity and the distance from  $P_{best_i}$  : to  $G_{best}$  is given as:

$$V_{i,m}^{t+1} = W * V_{i,m}^t + C_1^* rand() * (P_{best_i} - K_{i,m}^t) + C_2^* rand() * (G_{best_m} - K_{i,m}^t) \tag{11}$$

$$K_{i,m}^{t+1} = K_{i,m}^t + V_{i,m}^{t+1}, \text{ for } i = 1, 2, 3, \dots, n \tag{12}$$

where

$V_{i,m}^t$  : Velocity of particle at iteration  $i$

$W$  : Inertia weight factor

$C_1, C_2$  : Acceleration constant

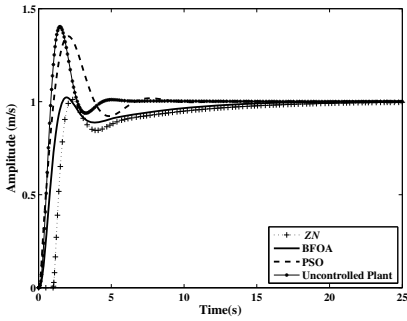
$rand()$  : Random no between 0 and 1

$K_{i,m}^t$  : Current position of the particle  $i$  at iteration.

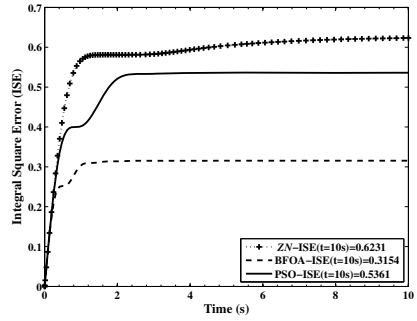


**Table 2.** Numerical Results

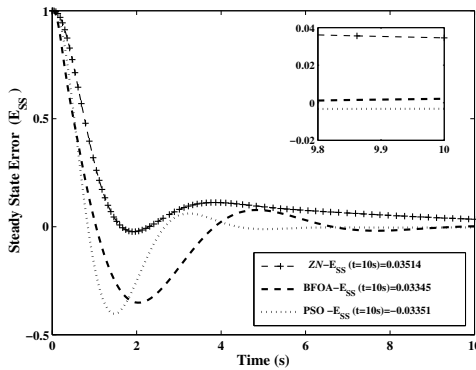
Parameters	Uncontrolled plant	ZN	BFOA	PSO
$K_P$	-	2.042	9.140	3.7915
$K_I$	-	0.24	12.42	3.4260
$K_D$	-	1.032	8.34	1.6192
Rise time	2.1702	0.6885	0.1135	0.3863
Settling time	33.53	14.6949	3.9597	4.9038
Settling min	1.8367	0.8451	0.9197	0.9171
Settling max	3.2135	1.0134	1.1660	1.2188
Overshoot	7.1158	1.3445	16.59	21.8866
Undershoot	0	0	0	0
Peak	3.2135	1.0134	1.1660	1.2188
Peak Time	4.1199	1.2777	0.2511	0.9927



**Fig. 3.** Output rotor response of the wind turbine system



**Fig. 4.** Simulation result of integral square error (ISE)



**Fig. 5.** Simulation result of steady state error

## 6 Conclusion

In this paper, the soft computing based control methodologies of the rotor response for horizontal axis wind turbine were presented. The optimization routines tried to optimize the gain parameters of PID controller ( $K_P$ ,  $K_I$  and  $K_D$ ) using the conventional *ZN* technique, the BFOA and the PSO algorithms. The optimized gain values of PID parameters with BFOA were obtained in smaller time than both the responses of the *ZN* and the PSO methods. The classical method can be used as a guess for obtaining the gain parameters of PID values in the BFOA and PSO algorithms. BFOA optimized gain parameters of PID controller performed superior in terms of the rise time and the settling time than the conventional method and PSO optimized ones. Also the integral square error and the steady state error associated with the BFOA based PID were lesser than the error obtained using *ZN* and PSO based PID controllers. The overall performance of BFOA optimized PID controller was superior than the two other compared algorithms.

**Acknowledgements.** The second author would like to thank the grant through Industrial Consultancy and Sponsored Research wing of Indian Institute of Technology, Madras for the grant through ‘New Faculty Seed Grant’.

## References

1. Slootweg, J.G., De Haan, S.W.H., Polinder, H., Kling, W.L.: General model for representing variable speed–wind turbines in power system dynamics simulations. *IEEE Tran. Power Syst.* 18(1), 144–151 (2003)
2. Salman, K.S., Teo, A.L.J.: Windmill modeling consideration and factors influencing the stability of a grid–connected wind power–based embedded generator. *IEEE Tran. Power Syst.* 18, 793–802 (2003)
3. Kim, D.H., Hoon, J.C.: Biologically inspired intelligent PID controller tuning for AVR systems. *Int. J. Control Automation Sys.* 4(5), 624–636 (2006)
4. Kim, D.H., Abraham, A.: A hybrid genetic algorithm and bacterial foraging approach for global optimization and robust tuning of PID controller with disturbance rejection. *Studies in Computational Intelligence* 75, 171–199 (2007)
5. Passino, K.M.: Bio mimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems Magazine* 17(08), 52–67 (2002)
6. Ying, S., Zengqiang, C., Zhuzhi, Y.: Adaptive constrained predictive PID controller via PSO. In: *Proceedings of the 26th Chinese Control Conference*, Zhangjiajie, Hunan, China, pp. 729–733 (2007)
7. Alrashidi, M.R.: A survey of PSO applications in electric power system. *IEEE Tran. Evolutionary Computation* 13(4), 913–918 (2009)
8. Aland, W.: Modern control design for flexible wind turbines. Technical Report NREL/TP-500-35816, NREL (2004)
9. Wright, A.D., Fingersh, L.J.: Advanced control design for wind Turbines: Control design, implementation and initial states. Technical report NREL/CP-500-36118, NREL (2008)

10. Hand, M.M.: Variable-speed wind turbine controller systematic design methodology: A comparison of nonlinear and linear model based design. Technical Report NREL/TP-500-25540, NREL (1999)
11. Leabi, S.K.: NN Self-Tuning pitch angle controller of wind power generation, M.S Thesis, University Of Technology, Baghdad, Iraq (2005)
12. Ogata, K.: Modern Control Systems Engineering. Prentice Hall, India (2010)
13. Nichita, C., Luca, D., Dakyo, B., Ceanga, E.: Large band simulation of the wind speed for real time wind turbine simulators. IEEE Tran. on Energy Conversion 17(4), 523–529 (2002)
14. Welfonder, E., Spanner, N.R.: Development and experimental identification of dynamics models for wind turbines. Control Engineering Practice 5(1), 63–73 (1997)

# Curve Fitting Using Coevolutionary Genetic Algorithms

Nejat A. Afshar, Mohsen Soryani, and Adel T. Rahmani

Department of Computer Engineering,  
Iran University of Science & Technology, Tehran, Iran  
n\_afshar@comp.iust.ac.ir, {soryani,rahmani}@iust.ac.ir

**Abstract.** Curve fitting has many applications in lots of domains. The literature is full of fitting methods which are suitable for specific kinds of problems. In this paper we introduce a more general method to cover more range of problems. Our goal is to fit some cubic Bezier curves to data points of any distribution and order. The curves should be good representatives of the points and be connected and smooth. These constraints and the big search space make the fitting process difficult. We use the good capabilities of the coevolutionary algorithms in large problem spaces to fit the curves to the clusters of the data. The data are clustered using hierarchical techniques before the fitting process.

**Keywords:** Curve fitting, Bezier curves, coevolutionary genetic algorithms, hierarchical clustering.

## 1 Introduction

Fitting of continuous and smooth curves to discrete data points is an essential task in many engineering problems. Geometric modeling, data analysis and image processing are some applications in which curve fitting is an important tool.

Many well established fitting methods are known, most of which are variants of least-squares techniques. These techniques perform well in problems with several defined parameters. For example in spline fitting, the process can be successful if the degree, knot positions, and the distribution of the data points are given and fixed. However in practice, these are not known, meanwhile these parameters influence the quality of the fitting result greatly. If they are not tuned properly, the fitting algorithm may have poor accuracy and the quality of the shapes will not be satisfactory. The complicated interdependence of the parameters and their influence on the fitting process is hard to control, but can be managed using evolutionary algorithms [1].

Evolutionary algorithms and their special case, genetic algorithms (GA) are a kind of stochastic search process inspired from Darwinian natural evolution, in which a population of candidate solutions are evolved. GAs are used in curve fitting [2-6] to avoid the complicated and unreliable process of finding the fitting parameters. [4] interpolates a cubic spline curve to data points by finding the knots using a genetic algorithm. Its goal is to minimize the curvature integral in order to reach an optimal shape. A real-coded genetic algorithm is developed in [2] to find good knots of a fitting spline. Data fitting with polygons to approximate an object curve is fundamental in pattern recognition, image processing and computer graphics. [5] reduces the integral

square error between the curve and the polygon using a GA. Fitting of univariate cubic splines to noisy data points is sought in [3] by applying a genetic search to find the number of the knots and balancing the interpolating and smoothness capabilities of the fitting splines. A GA fits some curves to functional data in [6].

Most of the related works in this context are suitable for specific kinds of problems and perform well in their appropriate domain. For example some of the developed methods in curve fitting are intended for functional data points and cannot be used for nonfunctional data. Some other fitting techniques can be applied for nonfunctional data points but the structure and the distribution of data should be of some specific type, e.g. closed shapes, connected regions and data without noise and outliers. Moreover the data points in most of the works in the literature should be in an ordered list and cannot be distributed randomly on a plane.

In this paper we propose a method for fitting some cubic spline curves to data points on a plane, independent of the distribution of the data. This method can be used for functional and nonfunctional data and the structure of data does not matter, i.e. data points can be closed or open shapes, connected or not connected and noise and outliers do not affect the fitting quality. First the data points are clustered in a way that every group of data is suitable for fitting a cubic spline. Next for each cluster a genetic algorithm is run to fit a Bezier spline to the cluster. These genetic algorithms are run simultaneously and cooperatively, consisting of a coevolutionary genetic algorithm. The final solution of the fitting algorithm is derived by combining the partial solutions of the GAs.

## 2 The Proposed Fitting Method

It is difficult to fit splines to data points of any distribution, since there are many unknown parameters. The number of the curves and the position of the knots are two important parameters to be detected, and many of the works on the literature focus on determination of these parameters [2, 7].

Our goal is to fit some smooth Bezier splines to data points on 2-dimensional space in a way that the distances between the curves and the points are minimum and the curves are good representatives of the data. We assume that the input data to be fitted are given on a plane and can be of arbitrary distribution. The set of data points consists of  $N$  points, having real values and can be written as

$$D = \{(x_i, y_i) \mid x_i, y_i \in \mathbb{R}\}, \quad i = 1, \dots, N. \quad (1)$$

The output of our algorithm is a set of  $M$  cubic Bezier splines, joined together smoothly as necessary. Each Bezier spline curve is in parametric form and can be defined by two knots and two control points and is written as (2)

$$B_j(t) = \sum_{i=0}^3 \binom{3}{i} t^i (1-t)^{3-i} P_{j,i}, \quad t \in [0, 1], \quad j = 1, \dots, M. \quad (2)$$

By setting proper values for knots and control points, appropriate fitting curves can be obtained.



The proposed method consists of two major phases. First the data points are grouped together using hierarchical clustering techniques, in order to prepare points for the fitting phase. Next a coevolutionary genetic algorithm is run; in which  $M$  population of individuals evolve together simultaneously and cooperatively to fit smooth Bezier curves to the clusters. The number of the clusters from the first phase is  $M$ , equal to the number of the output curves. Since the appropriate number of the curves is not known, by using hierarchical clustering and choosing a specific level of the resulting dendrogram, a good value for the number of the curves can be obtained. The location of the knots and control points is set by the coevolutionary process. By moving the knots between the adjacent clusters, an optimum position can be determined. The clustering phase and the coevolutionary process are described in the upcoming sections.

## 2.1 The Clustering Algorithm

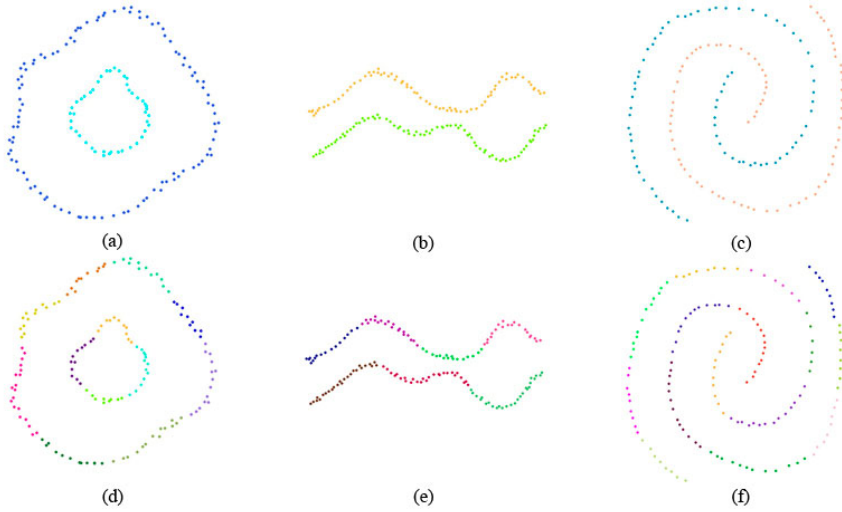
Clustering of data points is done in two steps. A single linkage clustering algorithm defines connected regions at the first step. At the second step, an average linkage clustering algorithm is run to partition each connected region to some related groups of data. These groups are later used in the genetic algorithm. The algorithm for the two step clustering is as follows.

- (1) Compute  $d_{avg}$ , the average distance between the points in  $D$ .
- (2) Create clusters  $C_{SL}$  of points in  $D$ , using single linkage algorithm and  $k*d_{avg}$  as minimum distance between clusters.
- (3) For each cluster  $C_i$  in  $C_{SL}$ :
  - a. Create clusters  $C_{AL}$  of points in  $C_i$ , using average linkage algorithm and a measure as minimum distance between clusters.
  - b. Add clusters in  $C_{AL}$  to  $C_{final}$ .
- (4) Output  $C_{final}$ .

The single linkage algorithm is responsible for putting different connected regions in different clusters. For this purpose a stop criterion is needed. If the distances between all the clusters are more than  $k*d_{avg}$ , the merging of the clusters stops.  $d_{avg}$  is a measure that shows the looseness of data points and can be computed as (3).

$$d_{avg} = \frac{1}{N} \sum_{i=1}^N \min_{j \neq i} d(D_i, D_j), \quad (3)$$

Each of the connected regions is clustered by an average linkage algorithm with a distance measure for stopping the algorithm like the previous step. The larger the measure, the smaller the number of the clusters and respectively the number of the curves. By setting these measures properly, the appropriate number of curves is achieved.



**Fig. 1.** Results of the clustering algorithm. Different clusters are shown in different colors. The results of the first step of the algorithm for the corresponding data points are shown in (a), (b) and (c). Final clusters are indicated in (d), (e) and (f).

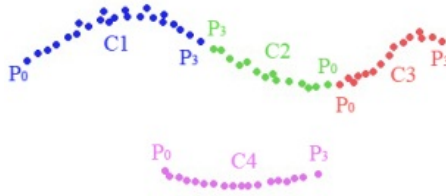
In order to improve the quality of fitting, noise and outliers can be omitted in this phase. This is done by removing the clusters smaller than a specific amount. Fig. 1 shows some results of the proposed clustering algorithm.

As indicated in Fig. 1 (a), (b) and (c), the single linkage algorithm detects the connected regions. These regions are defined with respect to the distance between the neighbor points. By this method data points are prepared for curve fitting easily. Fig. 1 (d), (e) and (f) show the final clusters that are ready for the coevolutionary process.

## 2.2 The Coevolutionary Process

The groups of data points created by the clustering algorithm are used in a coevolutionary process to yield a set of well-fitted splines. The coevolutionary genetic algorithm consists of  $M$  population of individuals, each one representing solutions for a specific cluster of points. Recall that  $M$  is the number of clusters created in the previous phase. These GAs work together to overcome the constraints of the overall problem. Each GA cooperates with the GAs in its neighborhood. The clusters with adjacent knots (start or end points) have corresponding neighbor GAs. Fig. 2 shows four clusters and their corresponding knots. As indicated, the clusters C1, C2 and C3 are in a connected region and their corresponding curves should be merged together. The cluster C2 has two neighbors, C1 and C3 have one neighbor and there is no neighbor for the cluster C4. The GA for C2 acts in direct cooperation with C1 and C3, independent of C4.

The farthest points in a cluster are considered as the two knots of it. Adjacent clusters are identified using the distance between their knots and a measure like the one



**Fig. 2.** Four clusters and their knots. Note that the start or end points of a cluster can be adjacent with even the start or end point of another cluster.

used in the previous section. If the distance between two knots of a cluster is less than  $k \cdot d_{avg}$ , then the clusters are considered as neighbors. The knots between adjacent clusters should be merged, so the mean point between them is assigned as the new knot.

After defining the neighbor clusters and setting their common knots, the data points are ready for the coevolutionary fitting algorithm. The proposed coevolutionary genetic algorithm is as follows.

- (1) Randomly generate  $M$  subpopulations of size  $N_{pop}$  and evaluate each Bezier curve in each subpopulation.
- (2) Keep the best individuals in each subpopulation.
- (3) For  $gen = 1$  to  $G$  do
- (4)     For  $i = 1$  to  $M$  do
- (5)         Select  $N_{pop}$  parents from subpopulation  $P_i$ , using tournament selection to breed offsprings.
- (6)         Perform crossover on parents with the probability of crossover rate and keep  $N_{pop}$  produced offsprings.
- (7)         Perform mutation on the bred offsprings with the probability of mutation rate.
- (8)         Evaluate each individual of the offsprings.
- (9)         Perform a selection on  $P_i$  and offsprings by tournament to get a new subpopulation  $\hat{P}_i$  of the same size as  $P_i$ .
- (10)        Replace  $P_i$  with  $\hat{P}_i$ .
- (11)        Keep the best curve in  $P_i$  and obtain appropriate curves for the neighbor GAs, so that the splines remain smooth.
- (12)        Correct the best curves in neighbor GAs and replace a part of their subpopulations with the revised curves.
- (13)        Select the best curves from each subpopulation to form the final Bezier splines and output it.

Some important parameters are the population size  $N_{pop}$ , the number of generations, and the crossover and mutation rates. These parameters can be set through trial and test.

Smoothness is a constraint that the GAs cope with it cooperatively. After a solution is found in a subpopulation, the solutions of the neighbor GAs should be updated in a way that the sequence of the joined curves remains smooth. These revised solutions are found in step (11) of the coevolutionary genetic algorithm. After that, a part of the subpopulations of the neighbor GAs are replaced with corrected solutions.

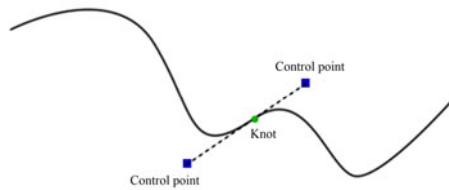
A spline is smooth if the tangent vectors of the piecewise curves are equal in common knots. This is indicated in Fig. 3. As indicated, the control points before and after a knot are on a direct line with the knot. This constraint can be used to obtain smooth curves. (4) shows the mathematical notation for the smoothness constraint. In this equation  $k$ ,  $c_1$  and  $c_2$  refer to the knot, the preceding and the subsequent control points respectively.

$$\frac{y_{c_1} - y_k}{x_{c_1} - x_k} = \frac{y_k - y_{c_2}}{x_k - x_{c_2}} \tag{4}$$

When a solution is found in a subpopulation, the solution of a neighbor subpopulation is updated by correcting the position of its control point near the common knot. The new position is a point on the line passing from the knot and the new control point of the found solution, having the same distance from the knot. This ensures that the final curves obtained by combining the partial solutions, remain smooth.

The Bezier curves are encoded in the chromosomes simply using their knots and control points. Since cubic splines are used here, the genotype of a chromosome consists of four genes. Each gene encodes a point, including the  $x$  and  $y$  characteristics of the point. The coding strategy is shown in Fig. 4. The start and end points are  $P_0$  and  $P_3$  respectively, and the two control points are  $P_1$  and  $P_2$ . The  $x$  and  $y$  in each gene have real values.

The distance of the points to the curves can be used as a good fitness measure. The fitness of a curve in subpopulation  $P_i$  can be computed as (5) and (6).



**Fig. 3.** An example of a smooth spline. Tangents of the two joined curves are equal at the knot

gene 1		gene 2		gene 3		gene 4	
$x_0$	$y_0$	$x_1$	$y_1$	$x_2$	$y_2$	$x_3$	$y_3$
$P_0$		$P_1$		$P_2$		$P_3$	

**Fig. 4.** The coding of Bezier curves in chromosomes

$$distance = \sum_{j=1}^{N_{C_i}} mind(c_{ij}, B_k) + \sum_{j=1}^{N_{C_L}} mind(c_{Lj}, B_k) + \sum_{j=1}^{N_{C_R}} mind(c_{Rj}, B_k), \quad (5)$$

$$k = i, L, R,$$

$$fitness = -(distance). \quad (6)$$

In (5),  $C_i$  is the  $i^{th}$  cluster,  $N_{C_i}$  is the number of the points in the  $i^{th}$  cluster,  $c_{ij}$  is the  $j^{th}$  point in the cluster  $C_i$ ,  $L$  and  $R$  are the index of the two neighbors of  $i$ ,  $B_k$  is the Bezier curve in  $k^{th}$  subpopulation and  $d$  is the distance between a point and a curve. The curves of neighbor subpopulations are first smoothed in according to the new curve. The distance between a point and a parametric Bezier curve is computed by combining Newton's method and quadratic minimization. This method is fast and robust. For a detailed description see [8].

As indicated in (6), the total distance of the points to the curves is negated and returned as the fitness, so that a less distance corresponds to a more fitness. Notice that the fitnesses of neighbor solutions influence the fitness of the current solution.

The GA operations are applied in the order of crossover, mutation and selection. First a set of individuals are selected for the recombination. Then a one-point crossover is applied to tuples of the parents with the probability  $p_c$ . This is done by selecting a random position in the genotype of the parents and then splitting both parents at this point and creating the two children by exchanging the tails. The mutation is done by adding to the current gene values of  $x$  and  $y$  an amount drawn randomly from a Gaussian distribution with mean zero and standard deviation  $\sigma$ . This takes place within probability of  $p_m$  for every gene. Tournament selection is used for parent and survivor selection. This selection mechanism is simple and fast to apply and the selection pressure can be easily controlled by varying the tournament size  $k$  [9].

### 3 Experimental Results

In this section, some test examples are provided to show the effectiveness of the proposed method. All of the data points here are randomly distributed on a plane, having real values in  $x$  and  $y$  dimensions and lie within a unit square. Some of the parameters of the CGA used through the experiments are shown in Table 1.

**Table 1.** Parameters of the CGA used throughout the experiments

Parameter	Value
Subpopulation size	10
Number of generations	50
Crossover rate	0.9
Mutation rate	0.2
Tournament size	2
Number of runs	30

The parameter  $k$  for the clustering algorithm is 3 and the standard deviation for the mutation is 2 in the experiments. 30 runs are performed for each of the examples.

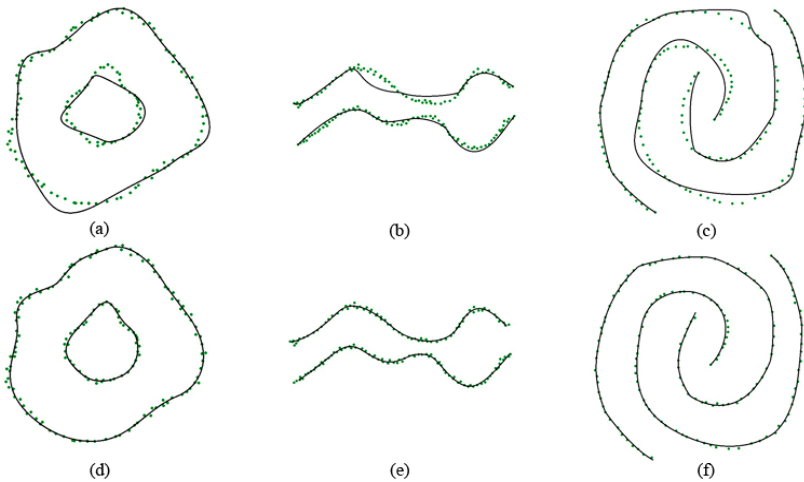
Fig. 5 shows the fitting results for the data points in Fig 1. As you can see, the points in Fig. 1 (c) are mingled together. Using the two step clustering algorithm, the data are prepared for the fitting algorithm. In the coevolutionary process, the positions of the knots are changed by genetic operators. As shown in Fig. 5, the final results of the CGA are a set of smooth curves that are good representatives of the data points.

Fig. 5 (a), (b) and (c) show the best curves in each subpopulation in the first generation of the best run of the coevolutionary process. As you can see the initial curves are approximately close to the final result. This is because of the good initialization of the individuals. The final results are shown in Fig. 5 (d), (e) and (f). These curves are the best solutions in each subpopulation in the 50th generation of the best run.

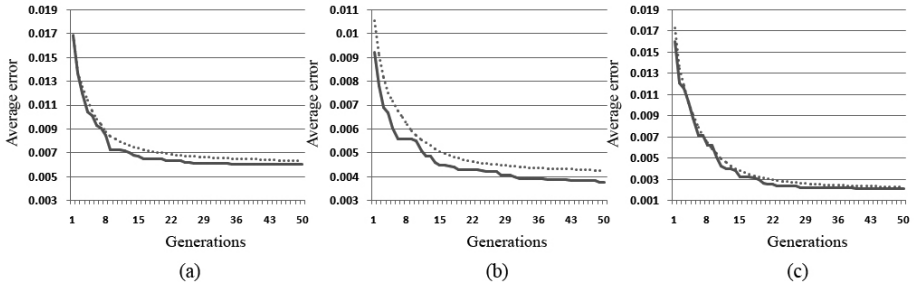
Fig. 6 (a), (b) and (c) show the average errors in each generation of the coevolutionary process. The average error is the average distance of the points to the curves. The dashed line in each diagram shows the average of the distances in 30 runs and the solid lines show the average distance in the best trial.

The experiments show that the coevolutionary process converges at about the 40th generation and only slight changes are made after that.

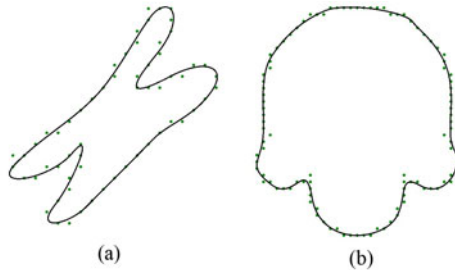
We have also compared our algorithm with some other curve fitting methods to show its effectiveness. The final results of our curve fitting method on two well-known data points are shown in Fig. 7. Table 2 presents a comparison with the proposed methods in [10-14].  $n_c$  is the number of the curves and ISE stands for integral square error and is the sum of the squared distance of all the points to the curves. As shown in Table 2 most of the methods do not consider smoothness. The advantage of our method is the smoothness of the curves with a relatively low error.



**Fig. 5.** Fitting results in the first and last generations. The final Bezier splines for the data points in Fig. 1 (a), (b) and (c) are presented in (d), (e) and (f) respectively.



**Fig. 6.** The average error for the generations of the CGA in the experiments. (a), (b) and (c) show the diagrams for the data points in Fig. 5 (a), (b) and (c). The dashed lines show the average results of 30 runs and the solid lines show the best trial in the runs.



**Fig. 7.** Results of the proposed method on two well-known data points. A chromosome-shaped curve and four semicircles are shown in (a) and (b) respectively.

**Table 2.** Results of the proposed algorithm with five other methods of curve approximation

Method	Chromosome			Semicircles		
	Smooth	$n_c$	ISE	Smooth	$n_c$	ISE
Pei and Horng [10]	Yes	15	0.0171	Yes	12	0.0094
Pei and Horng [11]	No	10	0.0083	No	4	0.0060
Horng and Li [12]	No	10	0.0074	No	4	0.0060
Sarkar et al. [13]	No	10	0.0074	No	4	0.0060
	No	11	0.0072	No	6	0.0056
	No	15	0.0061	No	12	0.0037
Pal et al. (TDLSA) [14]	No	5	0.0102	No	4	0.0060
Pal et al. (ESA) [14]	No	5	0.0062	No	4	0.0056
Proposed Method	Yes	10	0.0071	Yes	10	0.0076

## 4 Conclusions

In this paper, a new method for fitting a series of Bezier curves to data points is proposed. The points are placed randomly on a plane and there is no knowledge about

the distribution and the order of them. This method can be used for functional or nonfunctional data, data with closed or open shapes and can be scattered in different regions on the plane.

This method consists of two phases. First the data points are clustered using hierarchical clustering techniques. Next the prepared clusters are given to a coevolutionary process to fit a set of connected and smooth Bezier curves to them. The coevolutionary GAs can find good positions for the knots and the control points.

The experimental results show that the algorithm converges at about 40th generation. The final result is a set of continuous and smooth Bezier curves with a relatively low error. The major advantage of our method is that it can be used for data points with any distributions.

Our future work will be on more efficient clustering techniques to cope with more complex data and also using more capabilities of coevolutionary algorithms.

## References

1. Renner, G., Ekart, A.: Genetic algorithms in computer aided design. *Computer-Aided Design* 35, 709–726 (2003)
2. Yoshimoto, F., Harada, T., Yoshimoto, Y.: Data fitting with a spline using a real-coded genetic algorithm. *Computer-Aided Design* 35, 751–760 (2003)
3. Manela, M., Thornhill, N., Campbell, J.: Fitting spline functions to noisy data using a genetic algorithm. In: *Proceedings of the Fifth International Conference on Genetic Algorithms*, pp. 549–556. Morgan Kaufmann Publishers Inc. (1993)
4. Markus, A., Renner, G., Vancza, J.: Spline interpolation with genetic algorithms. In: *Proceedings of the International Conference on Shape Modeling and Applications*, pp. 47–54. IEEE (2002)
5. Yin, P.: Polygonal approximation using genetic algorithms. *Pattern Recognition*, 838–838 (1999)
6. Gulsen, M., Smith, A., Tate, D.: A genetic algorithm approach to curve fitting. *International Journal of Production Research* 33, 1911–1924 (1995)
7. Yang, H., Wang, W., Sun, J.: Control point adjustment for B-spline curve approximation. *Computer-Aided Design* 36, 639–652 (2004)
8. Wang, H., Kearney, J., Atkinson, K.: Robust and efficient computation of the closest point on a spline curve, pp. 397–406 (2002)
9. Eiben, A., Smith, J.: *Introduction to evolutionary computing*. Springer, Heidelberg (2003)
10. Pei, S.C., Horng, J.H.: Fitting digital curve using circular arcs. *Pattern Recognition* 28, 107–116 (1995)
11. Pei, S.C., Horng, J.H.: Optimum approximation of digital planar curves using circular arcs. *Pattern Recognition* 29, 383–388 (1996)
12. Horng, J.H., Li, J.T.: A dynamic programming approach for fitting digital planar curves with line segments and circular arcs. *Pattern Recognition Letters* 22, 183–197 (2001)
13. Sarkar, B., Singh, L.K., Sarkar, D.: Approximation of digital curves with line segments and circular arcs using genetic algorithms. *Pattern Recognition Letters* 24, 2585–2595 (2003)
14. Pal, S., Ganguly, P., Biswas, P.: Cubic Bézier approximation of a digitized curve. *Pattern Recognition* 40, 2730–2741 (2007)



# A Parallel Hybridization of Clonal Selection with Shuffled Frog Leaping Algorithm for Solving Global Optimization Problems (P-AISFLA)

Suresh Chittineni<sup>1</sup>, A.N.S. Pradeep<sup>2</sup>, G. Dinesh<sup>3</sup>,  
Suresh Chandra Satapathy<sup>1</sup>, and P.V.G.D. Prasad Reddy<sup>4</sup>

<sup>1</sup> Anil Neerukonda Institute of Technology and Sciences,  
Sangivalasa-531162, Visakhapatnam, Andhra Pradesh, India  
{sureshchittineni, sureshsatapathy}@gmail.com

<sup>2</sup> CISCO Systems, Bangalore  
pradeep.6174@gmail.com

<sup>3</sup> INFOSYS, Mysore  
dinesh.coolguy222@gmail.com

<sup>4</sup> Department of CS&SE, AU Engineering College,  
Visakhapatnam, Andhra Pradesh, India  
prasadreddy.vizag@gmail.com

**Abstract.** Shuffled frog leaping Algorithm (SFLA) is a new memetic, local search, population based, Parameter free, meta-heuristic algorithm that has emerged as one of the fast and robust algorithm with efficient global search capability. SFLA has the advantage of social behavior through the process of shuffling and leaping that helps for the infection of ideas. Clonal Selection Algorithms (CSA) are computational paradigms that belong to the computational intelligence family and is inspired by the biological immune system of the human body. CSA has the advantages of Innate and Adaptive Immunity mechanisms to antigenic stimulus that helps the cells to grow its population by the process of cloning whenever required. A hybrid algorithm is developed by utilizing the benefits of both social and immune mechanisms. This hybrid algorithm performs the parallel computation of social behavior based SFLA and Immune behavior based CSA to improve the ability to reach the global optimal solution with a faster and a rapid convergence rate. This paper compared the Conventional CLONALG and SFLA approaches with the proposed hybrid algorithm and tested on several standard benchmark functions. Experimental results show that the proposed hybrid approach significantly outperforms the existing CLONALG and SFLA approaches in terms of Mean optimal Solution, Success rate, Convergence Speed and Solution stability.

**Keywords:** Shuffled Frog Leaping Algorithm (SFLA), CLONALG and P-AISFLA.

## 1 Introduction

Swarm intelligence is a research hotspot in the artificial intelligence field. It mainly simulates the population behaviors of the complicate system in nature or society.

Shuffled Frog Leaping Algorithm (SFLA), inspired from the food hunting behavior of the frog and Clonal Selection Algorithm (CSA), inspired from the immune system of the human body are such examples of this field.

SFLA is a population based memetic algorithm that carries the process of memetic evolution in the form of infection of ideas from one individual to another in a local search. A shuffling strategy allows for the exchange of information between local searches to move toward a global optimum [1]. This algorithm is used to calculate the global optimal of complex optimization problems and proven to be one of robust and efficient algorithms [2].

On the other side Clonal Selection Algorithm (CSA) is an Artificial Immune System Technique (AIS) that is inspired by the functioning of Clonal selection theory of acquired immunity. Cloning and Mutation mechanisms in CSA help in attaining the global optimal. The algorithm provides two mechanisms for searching for the desired final pool of memory antibodies. The first is a local search provided via affinity maturation (hyper mutation) of cloned antibodies [3]. The second search mechanism provides a global scope and involves the insertion of randomly generated antibodies to be inserted into the population to further increase the diversity and provide a means for potentially escaping from local optimal [4].

Both these algorithms are powerful and efficient in finding the global optimal solution. SFLA achieves this with the help of social behavior and local search exploitation. On the other hand, CLONALG achieves this by performing the process of cloning and hyper mutation. A hybrid algorithm is developed by making use of the benefits of both social and immune mechanisms to speed up the rate of convergence and to achieve a stable solution. This hybrid algorithm performs the computation of both SFLA and CLONALG in a parallel fashion. This Hybrid evolutionary algorithm preserves the features of both SFLA and CLONALG and helps to ensure the global optimal in a faster way with a better converged mean solution. This hybrid algorithm is tested on various standard and well known bench mark optimization functions and found that, the proposed hybrid algorithm P-AISFLA outperforms both SFLA and CLONALG.

The remainder of the paper is organized as follows: section 2 describes the Shuffled Frog Leaping algorithm (SFLA) followed by a brief description of Clonal Selection Algorithm (CSA) in section 3. The section 4 briefs the important features of both SFLA and CLONALG in terms of Social and immune behavior perspectives and also describes our proposed Hybrid algorithm.

Section 5 gives further explanations, experimental analysis, simulation results to several benchmark problems and comparisons of our proposed algorithms with the conventional CLONALG and SFLA. Section 6 concludes our paper with some remarks and conclusions.

## 2 Shuffled Frog Leaping Algorithm (SFLA)

SFLA is a memetic algorithm inspired by the research of food hunting behavior of frogs. It is based on evolution of memes carried by the interactive frogs and by the global exchange of information among themselves. SFLA is a combination of both deterministic and random approaches [2].

The SFL algorithm progresses by transforming “frogs” in a memetic evolution. In this algorithm, frogs are seen as hosts for memes and described as a memetic vector. Each meme consists of a number of memotypes. The memotypes represent an idea in a manner similar to a gene representing a trait in a chromosome in a genetic algorithm. The frogs can communicate with each other, and can improve their memes by infecting (passing information) each other [5]. Improvement of memes results in changing an individual frog’s position by adjusting its leaping step size.

### **Steps in SFLA Algorithm**

The following are the steps involved [6] in the Shuffled Frog Leaping Algorithm (SFLA):

**Step 1. Random generation of frogs:** Initially generate the population of k frogs randomly within the feasible region.

**Step 2. Evaluate fitness:** The fitness values of each frog are calculated and then sort all the k frogs in ascending order according to the fitness value. Identify the global best frog  $X_g$  from the entire population.

**Step 3. Partition to memplexes:** Partition all the k frogs in to p memplexes, each containing q frogs such that  $k = p \times q$ . The partition is done in such a way that, the first frog should send to the first memplex, second one to the second memplex, similarly, the  $p^{th}$  frog to the  $p^{th}$  memplex and the  $(p+1)^{th}$  frog back to the first memplex.

**Step 4. Memetic Evolution:** In each memplex, identify the best frog  $X_b$  and the worst frog  $X_w$  to perform the process of memetic evolution. In this process, the worst frog  $X_w$  that is identified in each memplex should be improved as follows:

$$B_i = rand(.) \times (X_b - X_w) \quad (1)$$

$$New X_w = Old X_w + B_i \quad (2)$$

$$(-B_{max} \leq B_i \leq B_{max})$$

Where,  $rand(.)$  is a random number between 1 and 0, and  $B_{max}$  is the maximum allowed change in the frogs position. If this process produces a better frog, then it should be replaced by the newly generated frog. Otherwise,  $X_b$  is replaced by  $X_g$  in Equation (3) and the process is once again repeated. If non improvement becomes possible, in this case a random frog is generated which replaces the old frog.

**Step 5. Local exploration:** Repeat the process from step 3 for a specific number of iterations to improve the local search capability.

**Step 6. Shuffling:** Shuffling is the process of combining all the frogs in all the memplexes in to a single group.

**Step 7. Convergence check:** Return back to Step 2, if the termination criterion is not met, else stop.

## **3 Immune Optimization Algorithm (CLONALG)**

CLONALG is a population based Meta heuristic algorithm and its main search power relies on cloning operator and mutation operator. The Clonal Selection Algorithm (CSA)

reproduces only those individuals with high affinities and selects their improved and matures progenies. This strategy suggests that the algorithm performs a greedy search, where single members will be locally optimized and the newcomers yield a broader exploration of the search-space [4]. This characteristic makes the CSA very suitable for solving optimization tasks. The basic steps and working of Immune Optimization algorithm (CLONALG) is described as follows:

**Step 1. Antibody Pool (AB) Initialization**

Initially, an Antibody Pool (AB) is created with N antibodies choosing randomly in the search space. Antibodies are represented by the variables of the problem ( $ab_1, ab_2, \dots, ab_N$ ) which are potential solutions to the problem.

**Step 2. Antibody Selection**

For each Antibody ( $ab_i$ ), its corresponding affinity is determined. The affinity is determined by applying the problem’s fitness function to the antibody. These antibodies are then sorted according to the affinity evaluated. Select the n highest affinity antibodies.

**Step 3. Cloning**

Cloning is one of the key aspects in AIS. It is the process of producing similar populations of genetically identical individuals. The selected best n Antibodies will be replicated in proportionate to their antigenic affinity. The replicated antibodies i.e., Clones are maintained as a separate clone population C. The Number of Clones for each antibody can be calculated by the following equation:

$$N_c = \sum_{i=1}^n \text{round} \left( \frac{\beta \cdot N}{i} \right) \tag{3}$$

where  $N_c$  is the total number of clones generated,  $\beta$  is a multiplying factor, N is the size of Antibody Pool (AB) and  $\text{round}(\cdot)$  is the operator that rounds its argument towards the closest integer. Clone size of each selected antibody is represented by each term of this sum. Higher the affinity is, higher becomes the number of clones generated for the selected antibody [4].

**Step 4. Affinity Maturation**

The Clone Population C is now subjected to Mutation process, inversely proportional to its antigenic affinity measurement function. So, the lower affinity antibodies will undergo mutation process. This Maturation helps for low affinity antibodies to mutate more in order to improve their affinity values. If the mutated clone population results in a better affinity the earlier clone population affinity value, the recent mutated tuple values will be updated. So affinity maturation always results in better affinity antibodies. For Gray coding, Uniform mutation or Gaussian Mutation or Cauchy mutation can be used. In this Paper, Self-adaptive mutation using Gaussian distribution is used for making a search in the area surrounding the cell with high probability. And it has an outstanding ability of both local and global searching. The Gaussian mutation operator [7] can be described as follows:

**Gaussian mutation operator**

$$\begin{aligned} \theta_i^j &= \theta_i^j \times \exp \left( \tau_1 \times N(0,1) + \tau_2 \times N_j(0,1) \right) \\ ab_i^j &= ab_i^j + \theta_i^j \times N_j(0,1) \end{aligned} \tag{4}$$

$$\tau_1 = \left(\sqrt{2 \times \sqrt{D}}\right)^{-1}, \tau_2 = \left(\sqrt{2 \times D}\right)^{-1}$$

where  $\theta_i = \{\theta_1, \theta_2, \dots, \theta_D\}$ ,  $i=1, 2, 3 \dots N_c$ ,  $j = 1, 2, \dots, D$ ,

The parameter  $\theta_i^j$  is the mutation step of antibody  $ab_i^j$ ,  $\tau_1$  and  $\tau_2$  is the whole step and the individual step respectively.

Then, calculate the affinity of the mutated clones. The better affinity mutations are stimulated while the worse are restrained when antibody undergoes affinity mutation. The higher affinity values are taken for next generation while the Lower affinity antibodies will be deleted.

**Step 5. Antibody Restraint**

Inspired by the vertebrate immune system mechanism called antibodies restraint, includes the process of suppression and supplementation. This step keeps diversity and helps to find new solutions that correspond to new search regions by eliminating some percentage of the worst antibodies in the population and replacing with the randomly generated new antibodies. This helps the algorithm not to being trapped to local optimal. In antibodies restraint [8] [9], for each Iteration, the similar antibodies are removed and randomly generated antibodies are introduced in the removed antibody places of the Antibody Pool (AB).

**The pseudo code of Antibody restraint’s Redundancy Removal**

**Step1:** For every antibody, affinity is computed, and sorts the affinity values in descending order for antibody pool (AB). Get an antibody vector: Pop ( $ab_1, ab_2 \dots ab_N$ )  
 N is the total number of antibody in generation.

**Step2:** Set  $i=1, j=N$

```

Repeat:
    Checking: Is AB (i) and AB (j) Similar or not?
    If AB (i) is similar with AB (j)
        Delete AB (j) in population
    Else
        j=j-1
    End if
    i=i+1
Until i= j-1
    
```

**Step3:** If number of the antibodies < N

Add new randomly generated antibodies to antibody pool, AB.

**Step 6. Convergence Check:**

Repeat step 2 to 5, until the following conditions met:

- Algorithm undergoes a Specified number of Iterations.
- The optimal solutions in memory cells aren’t improved in a given generations.

**4 Hybrid Algorithm Combination of SFLA and CLONALG (P-AISFLA)**

Social behavior based SFLA and Immune behavior based CLONALG are combined together to evaluate a Hybrid Algorithm that has the advantages of both the conventional Algorithms.

According to this algorithm, initially consider  $2N$  population is randomly generated within the given range. Then determine their Corresponding fitness or affinity values. Then sort all the population in the ascending order basing on their Computed fitness or affinity values. Then divide the entire  $2N$  population in to two groups each with  $N$  population. Dividing should be done in such a way that, the best  $N$  Sorted population should go to one group and the remaining  $N$  sorted population should go to another group.

The social behavior consists of coordination process among the population through the exchange the information. So the best population will emerge if all the population

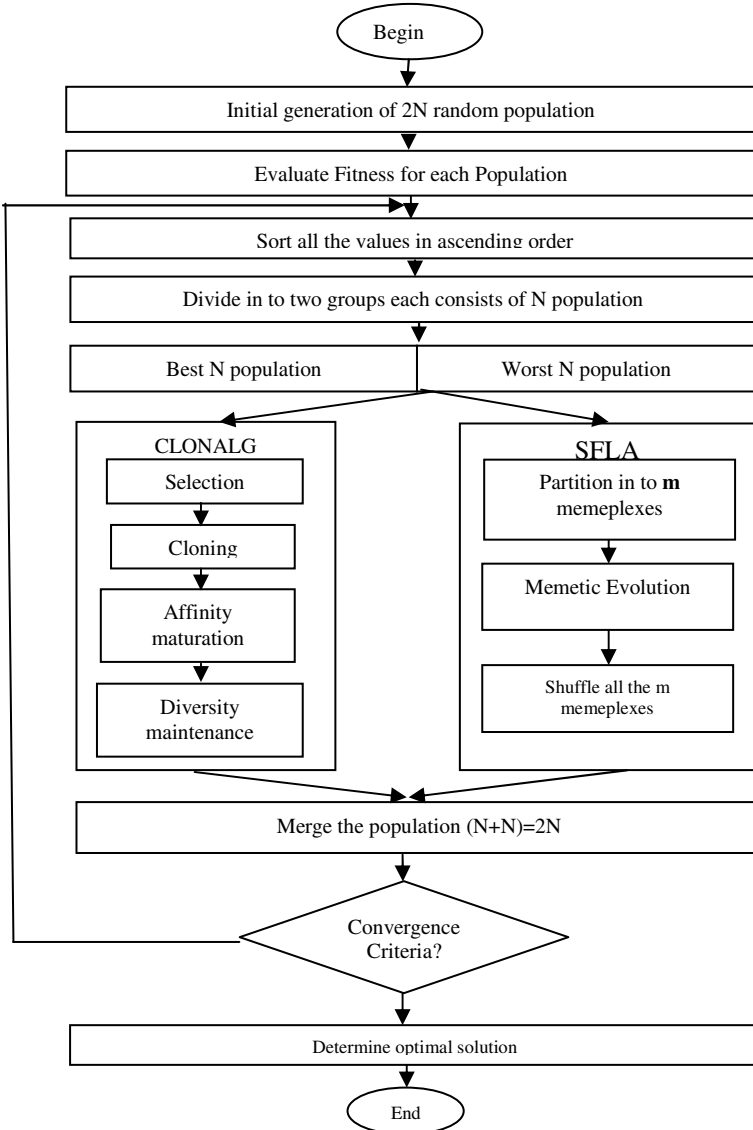


Fig. 1. Basic Flow chart of P-AISFLA

is sufficiently better where as the immune mechanism consists of mutation process that helps to improve the worst particle to the best particle.

Since SFLA consists of social behavior component, the worst group of population should be further processed by SFLA and the best group of population should be processed by CSA since CSA has immunity mechanisms.

After processing each group by the algorithm, the result of the two algorithms is merged again to become 2N population and the obtained 2N population should be repeated by the same above process. This whole of a process is repeated for specific number of iterations or until Convergence value is attained as the stopping criterion. The entire working of this Hybrid Evolutionary algorithm can be shown in fig.1.

## 5 Experiments and Results over Benchmark Functions

### A. Benchmark Functions for Simulation

A suite of standard benchmark functions [10] [11] [12] are taken into consideration to test the effectiveness and efficiency of the proposed approach P-AISFLA with the conventional CLONALG and SFLA. All these benchmark functions are meant to be minimized. The first function is unimodal with only one global minimum. The others are multimodal with a considerable number of local minima. Table 1 summarizes the expressions, Initializations and search ranges of the benchmark functions.

**Table 1.** Benchmark Test Functions of Dimension  $D$

	Function Name	Test Function	Domain Range
<b>F1</b>	Sphere	$\sum_{i=1}^D x_i^2$	$[-10,10]^D$
<b>F2</b>	Schwefel's Problem 2.22	$\sum_{i=1}^D  x_i  + \prod_{i=1}^D  x_i $	$[-10,10]^D$
<b>F3</b>	Rastrigin	$\sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-5.12,5.12]^D$
<b>F4</b>	Ackley	$-20 \exp \left( -0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2} \right) - \exp \left( \frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i) \right) + 20 + e$	$[-32,32]^D$
<b>F5</b>	Griewank	$\frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos \left( \frac{x_i}{\sqrt{i}} \right) + 1$	$[-600,600]^D$
<b>F6</b>	Zakharov	$\sum_{i=1}^N x_i^2 + (\sum_{i=1}^N 0.5i \cdot x_i^2) + (\sum_{i=1}^N 0.5i \cdot x_i^4)$	$[-10,10]^D$

Each of the Test Function has a Global Minimum Value of 0 [11] [12]

*B. Experimental Setup*

All the three algorithms were written in Matlab scripting language. The matlab codes were run on a computing Laptop machine at 1.8 GHz Intel processor. Each algorithm optimized each function 30 times independently.

The following simulation conditions are used for CLONALG:

- Initial Antibody Pool Size,  $AB = 20$
- Clonal Multiplying factor's range,  $\beta=[0.5-1]$
- Mutation type used: Gaussian
- Gaussian mutation probability  $P_{mg} = 0.01$
- Percentage of Suppression  $P_{sup} = 0.2$
- Number of Iterations=1000
- Number of Dimensions taken for each Benchmark Function=15

The following simulation conditions are used for SFL Algorithm

- Initial Population,  $N_{pop} = 20$
- Number of Memplexes,  $M= 2$
- Max. Local search iterations 10
- Number of Main Iterations=1000
- Number of Dimensions taken for each Benchmark Function=15

The number of generations used for CLONALG, SFLA and the proposed hybrid approach is P-AISFLA was 30. In each of the 30 computations, CLONALG, SFLA and P-AISFLA employed different initial populations randomly generated in the problem's search space. If the algorithm attained a solution with an optimal functional value equal to the criterion value  $\Theta$ , the solution is regarded as a success (Successful optimal Solution) or otherwise fails. The Success criterion value,  $\Theta$  chosen for individual benchmark function is tabulated in Table 2.

**Table 2.**  $\Theta$  values for Benchmark functions

Function Name	F1	F2	F3	F4	F5	F6
Success Criterion value $\Theta$	0.01	0.01	0.12	0.1	0.0001	0.001

*C. Benchmark Results and Analysis*

In this section, the results for the 6 benchmark test functions are given to show the merits of the proposed P-AISFLA. The experimental results in terms of the Mean Optimal Value, Standard deviation to measure the stability of optimal value, Success rate and the number of function evaluations are summarized in Tables 3- 8 and Figs. 2-7.

In order to further demonstrate the performance of P-AISFLA over CLONALG and SFLA, the evolutionary curves of optimizing each function by the 3 algorithms are given. The number of objective function evaluations is used to evaluate the convergence speed. Since the 3 algorithms are all randomized stochastic algorithms, it is not reasonable to compare their performances by optimizing these functions only one time. To reduce the stochastic or random influences, each algorithm optimized each function for 30 times, and the average optimal function value was calculated by:



$$MOV = \frac{\sum_{i=1}^{30} F(i)}{30} \tag{5}$$

where: MOV is Mean Optimal Value, F(i) is the Optimal value achieved in i<sup>th</sup> generation. The evolutionary curves of average optimal function values against number of iterations for each function are illustrated in Figs. 2–7.

A Successful Optimal Solution represents that it converges to the global optimum. Otherwise, the algorithms were regarded as getting stuck or trapped at the local optimal. Success rate, SR is the percentage of summation of number of times successful optimal solutions are obtained to the number of generations for a specific benchmark function.

Tables 3–8 illustrates that the accurate and high precision values cannot be obtained by CLONALG. When the function to be optimized is having many local optimal solutions with multiple modes, the algorithm is sometimes trapping or sticking at the local optimal. But, due to local search operator in SFLA, it is producing more accurate and global optimal solutions than CLONALG. The “SR” value for majority of the functions is high for SFLA than CLONALG.

Due to Cloning, mutation and local search operators available in P-AISFLA, the global search capability of it is dominant in comparison with CLONALG and SFLA and its convergence speed is predominantly faster than CLONALG and SFLA in terms of function evaluations for majority of the functions. It can be observed that the proposed approach possesses a relatively higher Success Rate SR than both CLONALG and SFLA implying that P-AISFLA has nice global search performance to avoid local optimal and the optimal solution gained by it is the most accurate. The experimental results are given in Tables 3–8.

**Table 3.** Optimization Results for Sphere function

F1	MOV	STD	FE	SR
CLONALG	1.147e-003	2.811 e-003	34446	70
SFLA	7.157e-009	3.6606e-009	11312	100
P-AISFLA	<b>7.495e-017</b>	<b>1.4838e-16</b>	<b>1288</b>	<b>100</b>

**Table 4.** Optimization Results for Schewefel’s F2 function

F2	MOV	STD	FE	SR
CLONALG	1.341e-001	2.087e-002	35000	30
SFLA	1.4542e-004	5.2831e-005	11312	80
P-AISFLA	<b>2.4206e-011</b>	<b>3.0946e-11</b>	<b>2419</b>	<b>95</b>

**Table 5.** Optimization Results for Rastrigin function

F3	MOV	STD	FE	SR
CLONALG	4.0898	7.026e-001	34550	25
SFLA	1.2097	2.290e-001	30316	55
P-AISFLA	<b>7.709e-002</b>	<b>1.123e-002</b>	<b>21000</b>	<b>90</b>

**Table 6.** Optimization Results for Ackley function

F4	MOV	STD	FE	SR
CLONALG	3.0761e-001	1.371e-001	40000	80
SFLA	6.124e-002	1.087e-002	32500	100
<b>P-AISFLA</b>	<b>2.0263e-004</b>	<b>3.2010e-004</b>	<b>15782</b>	<b>100</b>

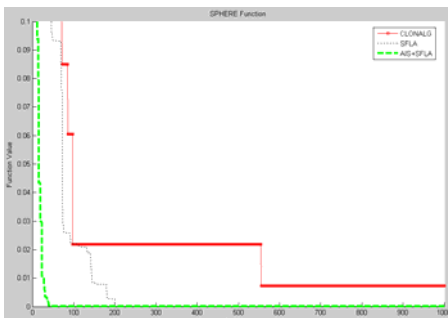
**Table 7.** Optimization Results for Griewank function

F5	MOV	STD	FE	SR
CLONALG	1.71e-004	2.101e-004	17442	90
SFLA	3.459e-003	1.473e-003	19473	75
<b>P-AISFLA</b>	<b>1.2107e-007</b>	<b>2.8321e-006</b>	<b>2688</b>	<b>100</b>

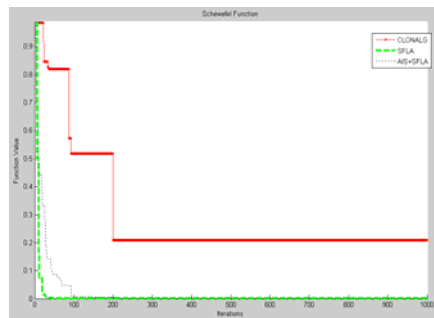
**Table 8.** Optimization Results for Zakharov function

F6	MOV	STD	FE	SR
CLONALG	5.49e-002	1.5e-002	32850	45
SFLA	4.9234e-008	3.768e-008	8616	100
<b>P-AISFLA</b>	<b>1.5127e-012</b>	<b>2.4749e-011</b>	<b>1583</b>	<b>100</b>

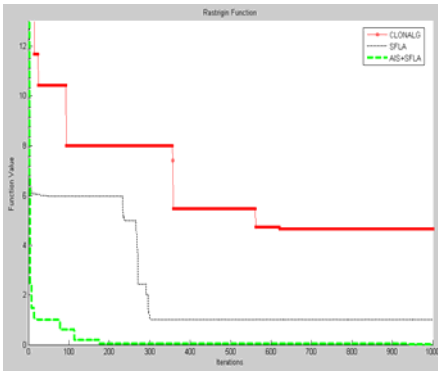
In these tables, ‘MOV’ is the Mean optimal value obtained by these algorithms; ‘STD’ is the Standard Deviation for the Mean optimal solution; ‘SR’ is the percentage of success for converging to the global optimum at 30 independent runs, and it reflect the global search performance of each individual algorithm. ‘FE’ is the average number of objective function evaluations required by these algorithms to attain the criterion values for the functions; ‘FE’ reflects the convergence speed of an algorithm.



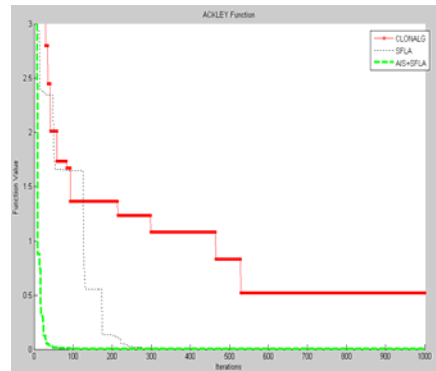
**Fig. 2.** Sphere Function



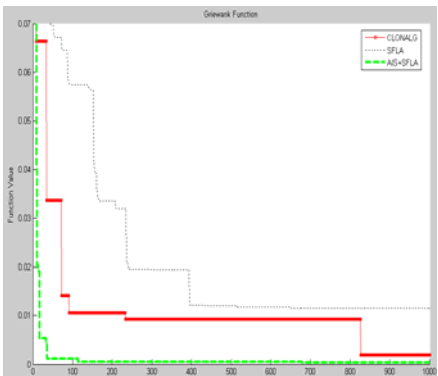
**Fig. 3.** Schewefel's Function



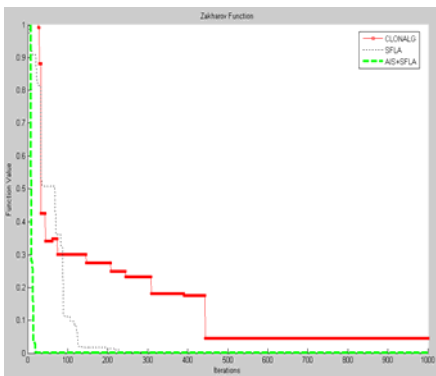
**Fig. 4.** Rastrigin function



**Fig. 5.** Ackley function



**Fig. 6.** Griewank Function



**Fig. 7.** Zakharov function

## 6 Conclusions

In this paper, we have proposed a novel and hybrid approach, P-AISFLA that utilizes the benefits of both cloning operator and local search operator in a parallel fashion. The local search operator and Mutation operators are used for improving local and global search capabilities. The parallel search mechanism is utilized to improve the performance of search efficiency of the algorithm. On solving a suite of benchmark functions, P-AISFLA performs better than both CLONALG and SFLA in terms of Mean optimal solution values, stability of the optimal solution, number of function evaluations and success rate. Simulation results on Standard benchmark problems demonstrate that the proposed method is a useful technique to solve complex and higher dimensional optimization problems.

## References

- [1] Elbeltagi, E., Hegazy, T., Grierson, D.: Comparison among five evolutionary-based optimization algorithms. *Advanced Engineering Informatics* 19), 43–53 (2005)
- [2] Eusuff, M.M., Lansey, K.E., Pasha, F.: Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization. *Engineering Optimization* 38(2), 129–154 (2006)
- [3] de Castro, L.N., Von Zuben, F.J.: *Learning and Optimization Using the Clonal Selection Principle*. *IEEE Trans. on Evolutionary Computation* 6(3), 239–251 (2002)
- [4] Castro, D., Timmis, J.: *An Introduction to Artificial Immune Systems: A New Computational Intelligence Paradigm*. Springer, Heidelberg (2002)
- [5] Liong, S.-Y., Atiquzzaman, M.: Optimal design of water distribution network using shuffled complex evolution. *J. Inst. Eng.* 44(1), 93–107 (2004)
- [6] Zhang, X., Hu, X., Cui, G., Wang, Y., Niu, Y.: An Improved Shuffled Frog Leaping Algorithm with Cognitive Behavior. In: *Proceedings of the 7th World Congress on Intelligent Control and Automation, Chongqing, China, June 25-27 (2008)*
- [7] Cortes, Coello, C.: *Handling Constraints in Global Optimization using an Artificial Immune System*
- [8] Timmis, J., Edmonds, C., Kelsey, P.: Assessing the Performance of Two Immune Inspired Algorithms and a Hybrid Genetic Algorithm for Function Optimisation. In: *Proceedings of the Congress on Evolutionary Computation*, pp. 1044–1051 (2004)
- [9] Pan, L., Fu, Z.: A Clonal Selection Algorithm for Open Vehicle Routing Problem. In: *Proceedings of Third International Conference on Genetic and Evolutionary Computing (2009)*
- [10] Zuo, X.Q., Fan, Y.S.: A chaos search immune algorithm with its application to neuro-fuzzy controller design. *Chaos, Solitons and Fractals* 30, 94–109 (2006)
- [11] Suganthan, Baskar, S.: Comprehensive Learning Particle Swarm Optimizer for Global Optimization of Multimodal Functions. *IEEE Trans. on Evolutionary Computation* 10(3) (June 2006)
- [12] Ling, S.H., Iu, C.: Hybrid Particle Swarm Optimization With Wavelet Mutation and Its Industrial Applications. *IEEE Tran. on Systems, Man and Cybernetics-Part B: Cybernetics* 38(3) (June 2008)

# Non-uniform Circular-Shaped Antenna Array Design and Synthesis - A Multi-Objective Approach

Saurav Ghosh<sup>1</sup>, Subhrajit Roy<sup>1</sup>, Sk. Minhazul Islam<sup>1</sup>, Shizheng Zhao<sup>2</sup>,  
Ponnuthurai Nagaratnam Suganthan<sup>2</sup>, and Swagatam Das<sup>1</sup>

<sup>1</sup>Dept. of Electronics and Telecommunication Engg.,  
Jadavpur University, Kolkata 700 032, India

<sup>2</sup>Dept. of Electronics and Electrical Engg.,  
Nanyang Technological University  
saurav\_online@yahoo.in,  
{roy.subhrajit20,skminha.isl}@gmail.com,  
ZH0047NG@e.ntu.edu.sg, epnsugan@ntu.edu.sg,  
swagatamdas19@yahoo.co.in

**Abstract.** Design of non-uniform circular antenna arrays is one of the important optimization problems in electromagnetic domain. While designing a non-uniform circular array the goal of the designer is to achieve minimum side lobe levels with maximum directivity. In contrast to the single-objective methods that attempt to minimize a weighted sum of the four objectives considered here, in this article we consider these as four distinct objectives that are to be optimized simultaneously in a multi-objective (MO) framework using one of the best known Multi-Objective Evolutionary Algorithms (MOEAs) called NSGA-II. This MO approach provides greater flexibility in design by producing a set of final solutions with different trade-offs among the four objective from which the designer can choose one as per requirements. To the best of our knowledge, other than the single objective approaches, no MOEA has been applied to design a non-uniform circular array before. Simulations have been conducted to show that the best compromise solution obtained by NSGA-II is far better than the best results achieved by the single objective approaches by using the differential evolution (DE) algorithm and the Particle Swarm Optimization (PSO) algorithm.

## 1 Introduction

Antenna arrays have an important role in detecting and processing signals arriving from different directions. Nowadays, antenna arrays [16] are preferred over single element antenna due to the latter's limitations in directivity and bandwidth. Design of antenna arrays overcomes such defects by associating each antenna elements in various electrical and geometrical configurations. The basic need for design of antenna array structure is to find out the positions of array elements that produce a radiation pattern in a whole that closely matches the desired pattern [1]. Recently synthesis of linear array elements separated in a non-linear fashion became immensely popular among the researchers working in electromagnetic domain.

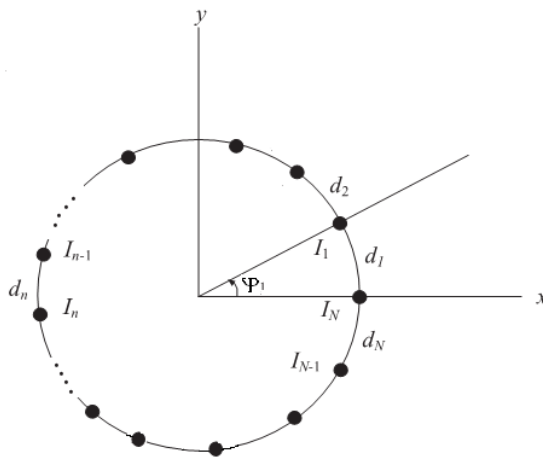
However, the researchers have also started to show special interest in the design of antenna arrays of different shapes such as the circular one which now find immense applications in sonar, radar, mobile, and commercial satellite communication systems [3, 4]. Till now, several single-objective meta-heuristic algorithms such as real-coded Genetic Algorithm (GA) [5], PSO [6], DE [7] have been applied to design non-uniform circular antenna arrays with minimum SLL, maximum directivity and null control. Design of circular antenna arrays requires optimization of four objectives. In the single objective approach, these four separate objectives are combined through a weighted linear sum into a single objective function. However, the weighted sum method is not appropriate and the solution obtained depends hugely on the relative values of the weights. In this article we have tried to solve the non-uniform circular antenna array problem with a multi-objective (MO) approach using NSGA-II [8]. Already some MO-based approaches [15, 17] have been made successfully in the field of antenna design. Unlike single-objective optimization techniques that return only a single best solution, the MOEAs [18] generate a set of non-dominated solutions (the Pareto optimal set). In order to validate the MO design method, we undertake a comparative study over one instantiation of the design problem comprising 8 element circular array. The best compromise solution obtained by NSGA-II is compared with two single-objective algorithms, namely DE [9] and PSO [10]. The comparison indicates that NSGA-II yields much better solutions as compared to DE and PSO demonstrating the effectiveness of multi-objective approach over single-objective approaches.

## 2 General Description of NSGA-II

NSGA-II [8] is non-domination based genetic algorithm for multi-objective optimization which incorporates elitism and no sharing parameter needs to be chosen *a priori*. The population is initialized as usual. Once the population is initialized the population is sorted based on non-domination into each front. The first front being completely non-dominant set in the current population and the second front being dominated by the individuals in the first front only and the front goes so on. Each individual in each front are assigned rank (fitness) values or based on front in which they belong to. Individuals in first front are given a fitness value of 1 and individuals in second are assigned fitness value as 2 and so on. In addition to fitness value a new parameter called *crowding distance* is calculated for each individual. The crowding distance is a measure of how close an individual is to its neighbours. Large average crowding distance will result in better diversity in the population. Parents are selected from the population by using binary tournament selection based on the rank and crowding distance. An individual is selected in the rank is lesser than the other or if crowding distance is greater than the other. The selected population generates off-springs from crossover and mutation operators. The population with the current population and current off-springs is sorted again based on non-domination and only the best  $N$  individuals are selected, where  $N$  is the population size. The selection is based on rank and the on crowding distance on the last front.

### 3 Multi-Objective Formulation of the Design Problem

The circular antenna array, being considered here, is non-uniform and planar, i.e. the elements are non-uniformly spaced on a circle of radius  $r$  in the  $x - y$  plane, as depicted in Figure 1. The elements comprising the circular antenna array are assumed to be isotropic sources. Now to calculate the array factor, we need to know the parameters of the array. The required parameters are the excitation current amplitude ( $I_n$ ), phase ( $\beta_n$ ), the angular position  $\varphi_n$  of the  $n$ -th element, and the circular arc separation between any two adjacent elements ( $d_n$ -the distance between elements  $n$  and  $n - 1$ ).



**Fig. 1.** Geometry of a non-uniform circular antenna array element with  $N$  isotropic radiators or elements

The expression for the array factor in the  $x - y$  plane can be represented as:

$$AF(\varphi) = \sum_{n=1}^N I_n e^{j(kr \cdot \cos(\varphi - \varphi_n) + \beta_n)} \tag{1}$$

Now  $kr$  and  $\varphi_n$  can be formulated as

$$\left. \begin{aligned} kr &= \frac{2\pi r}{\lambda} = \sum_{i=1}^N d_i, \\ \varphi_n &= \frac{2\pi}{kr} \sum_{i=1}^n d_i, \end{aligned} \right\} \tag{2}$$

In this article, our goal is to synthesize a circular antenna array with minimum side lobes level (SLL) and maximum directivity. The maximum side lobe level along with the average side lobe level is to be included in the fitness function for the directivity purposes. The following objective functions can be formulated in a mathematical form as shown below to satisfy these requirements:

$$f_{NU} = |AF(\varphi_{NULL1})| + |AF(\varphi_{NULL2})| \tag{3}$$

$$f_{SLA} = \frac{1}{\pi + \varphi_{NULL1}} \int_{-\pi}^{\varphi_{NULL1}} |AF(\varphi)| d\varphi + \frac{1}{\pi - \varphi_{NULL2}} \int_{\varphi_{NULL2}}^{\pi} |AF(\varphi)| d\varphi \tag{4}$$

$$f_{MSL} = |AF(\varphi_{MSLL1})| + |AF(\varphi_{MSLL2})| \tag{5}$$

where  $\varphi_{NULL1}$  and  $\varphi_{NULL2}$  are the two null angles,  $\varphi_{MSLL1}$  is the angle where the maximum side lobe level (SLL) is obtained in the lower band  $[-\pi, \varphi_{NULL1}]$  and,  $\varphi_{MSLL2}$  is the angle where the maximum side lobe level (SLL) is obtained in the lower band  $[\varphi_{NULL2}, \pi]$ . In addition, to satisfy practical spatial requirements we incorporate another objective function which is the array circumference (since, planar array) that can be mathematically formulated as:

$$f_D = \sum_{i=1}^N d_i, \tag{6}$$

where,  $d_i$ 's is the distance between element  $i$  and  $i-1$ ,  $i = 1, 2, 3, \dots, N$ . The equations 3, 4, 5 and 6 define the four objective functions to be optimized in a multi-objective approach, i.e. by a MOEA. While diminishing the array size one needs to keep in mind that the antenna directivity also decreases. An MOEA will allow us to find trade-off solutions between the four objectives shown above in order to achieve minimum side lobe level and maximum directivity with moderate circumference size. So an MOEA will provide greater flexibility in designing a non-uniform circular antenna array because a single-objective EA gives us only one solution in one run which might not completely satisfy the designer's needs.



## 4 Simulations and Results

One instantiation of the design problem, namely 8 element non-uniform circular antenna array is solved in a MO framework by using the NSGA-II algorithm. We compare the best compromise solution obtained by NSGA-II with the best results achieved by the single-objective optimization techniques, namely DE and PSO. This DE variant is known as DE/rand/1/bin and is the most popular one in DE literature [11]. The PSO variant used here is the original PSO algorithm as given in [10]. In case of single-objective optimization algorithms the objective function is the weighted sum of the four objectives considered here. In what follows we report the best results obtained from a set of 50 independent runs of NSGA-II and its single-objective competitors, where each run for each algorithm is continued up to 10000 Function Evaluations (FEs). Note that for NSGA-II we extract the best compromise solution obtained with the fuzzy membership function based method outlined in [13]. For NSGA-II and the contestant algorithms we employ the best suited parametric set-up chosen with guidelines from their respective literatures. To satisfy the requirements of practical considerations, current amplitudes are normalized with maximum value being set equal to 1.

Achieving the right balance between null control and side lobe level suppression has always been an issue in antenna design. So in Table 1 we provide the performance results of the two objectives  $f_{NU}$  and  $f_{SLA}$  for NSGA-II and the competitor single-objective algorithms DE and PSO over the single design instance which basically determines the balance between side lobe level suppression and null control. Table 2 presents the best values (out of 50 independent runs) of the two important factors which the designer is concerned of – Maximum SLL (in decibels), and Directivity (in decibels) obtained with the NSGA-II and the single-objective meta-heuristics DE and PSO. Figure 2 delineates the radiation patterns of the non-uniform circular antenna arrays generated by NSGA-II and all the other competitor algorithms for 8 element array.

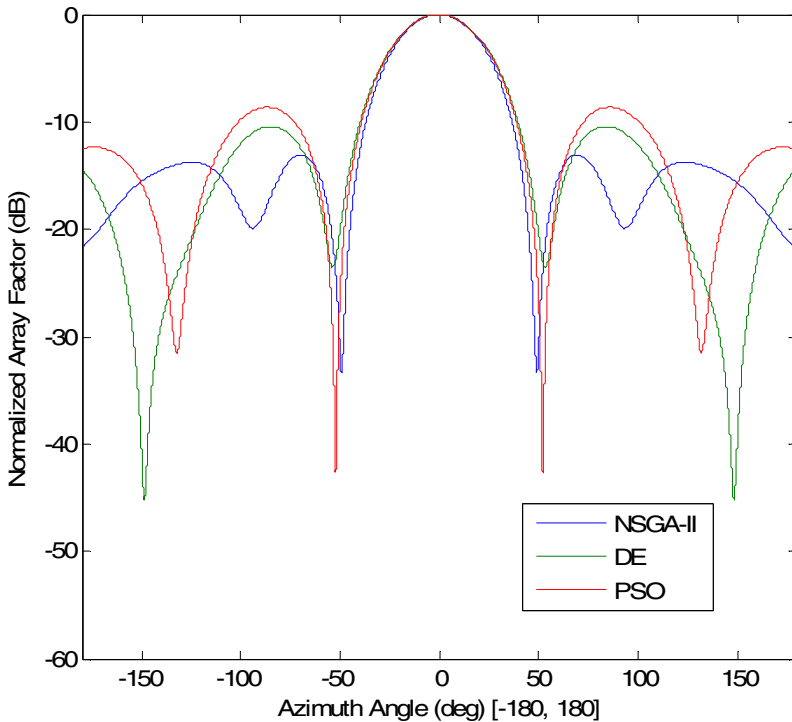
The overall performance improvement regarding the circular antenna design can be derived from Table 2. A close scrutiny of Table 2 reveals that the NSGA-II yields much better values of two figures of merit – the maximum SLL (in dB) and the directivity (in dB) in comparison to the competitor algorithms. From Figure 2, it is clearly seen that the normalized array factor for the optimization parameter values achieved with NSGA-II has better side-lobe suppression than those of DE and PSO.

**Table 1.** Design objectives achieved with the three algorithms

Number of Elements	Algorithm	$f_{NU}$	$f_{SLA}$
8	NSGA-II	<b>0.7360</b>	<b>0.2707</b>
	PSO	0.8462	0.4315
	DE	0.8454	0.3820

**Table 2.** Design figures of merit obtained in the best (out of 50) run of the three algorithms on the 8 element array design instance

Number of Elements	Algorithm	Maximum SLL (in dB)	Directivity (in dB)
8	NSGA-II	<b>-13.1</b>	<b>11.7276</b>
	PSO	-9.2	9.6936
	DE	-10.7	10.2401



**Fig. 2.** Normalized Radiation Patterns obtained for 8 element circular antenna arrays using NSGA-II, DE and PSO

## 5 Conclusion

Synthesizing non-uniform circular antenna arrays with minimum SLL and maximum directivity has emerged as a popular as well as challenging optimization problem among the researches in electromagnetic domain. Contrary to the inefficient meta-heuristic single-objective approaches made before, in this article, we propose a multi-objective (MO) framework in which the four objectives associated with the circular arrays are optimized simultaneously with the help of NSGA-II algorithm. This MO framework provides immense advantages to the designer as the MOEAs generate the Pareto optimal set from which the designer can choose a desired solution and helps in finding the right balance between the four objectives. Our simulation experiments indicated that the best compromise solution obtained by NSGA-II algorithm could comfortably outperform the best results obtained with the traditional single-objective DE and PSO algorithms over the 8 element array design problem demonstrating the efficiency as well as effectiveness of the MO framework over the single objective one. NSGA-II is also able to achieve minimum SLL and maximum directivity among the competitors.

Future research may focus on designing other antenna array geometries as well as concentric arrays with the help of MO framework, in which different components of the cost function are treated as a multi-objective optimization problem. Other recent MO algorithms like 2LB-MOPSO [12], MODE-fast-sorting [14], MOEA/D-DE [2] as well as additional novel MO algorithms will be applied for solving these problems in order to comparatively evaluate different algorithms.

## References

1. Rahmat-Samii, Y., Michielssen, E. (eds.): *Electromagnetic Optimization by Genetic Algorithms*. Wiley, New York (1999)
2. Zhang, Q., Li, H.: MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Trans. Evolutionary Computation*, 712–731 (2007)
3. Gurel, L., Ergul, O.: Design and simulation of circular arrays of trapezoidal-tooth log-periodic antennas via genetic optimization. *Progress in Electromagnetics Research, PIER* 85, 243–260 (2008)
4. Dessouky, M., Sharshar, H., Albagory, Y.: A novel tapered beamforming window for uniform concentric circular arrays. *Journal of Electromagnetic Waves and Applications* 20(14), 2077–2089 (2006)
5. Panduro, M., Mendez, A.L., Dominguez, R., Romero, G.: Design of non-uniform circular antenna arrays for side lobe reduction using the method of genetic algorithms. *Int. J. Electron. Commun (AEU)* 60, 713–717 (2006)
6. Shihab, M., Najjar, Y., Dib, N., Khodier, M.: Design of non-uniform circular antenna arrays using particle swarm optimization. *Journal of Electrical Engineering* 59(4), 216–220 (2008)
7. Panduro, M.A., Brizuela, C.A., Balderas, L.I., Acosta, D.A.: A comparison of genetic algorithms, particle swarm optimization and the differential evolution method for the design of scannable circular antenna arrays. *Progress in Electromagnetics Research B* 13, 171–186 (2009)

8. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6(2), 182–197 (2002)
9. Das, S., Suganthan, P.N.: Differential Evolution: A Survey of the State-of-the-Art. *IEEE Transactions on Evolutionary Computation* 15(1), 4–31 (2011), doi:10.1109/TEVC.2010.2059031
10. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: *Proceedings of the IEEE International Conference Neural Networks*, vol. 4, pp. 1942–1948 (1995)
11. Price, K., Storn, R., Lampinen, J.: *Differential Evolution - A Practical Approach to Global Optimization*. Springer, Berlin (2005)
12. Zhao, S.Z., Suganthan, P.N.: Two-lbests Based Multi-objective Particle Swarm Optimizer. *Engineering Optimization* 43(1), 1–17 (2011), doi:10.1080/03052151003686716
13. Tapia, C.G., Murtagh, B.A.: Interactive fuzzy programming with preference criteria in multiobjective decision making. *Comput. Oper. Res.* 18, 307–316 (1991)
14. Qu, B.Y., Suganthan, P.N.: Multi-Objective Evolutionary Algorithms based on the Summation of Normalized Objectives and Diversified Selection. *Information Sciences* 180(17), 3170–3181 (2010)
15. Pal, S., Qu, B.Y., Das, S., Suganthan, P.N.: Optimal Synthesis of Linear Antenna Arrays with Multi-objective Differential Evolution. *Progress in Electromagnetics Research, PIER B* 21, 87–111 (2010)
16. Chandran, S. (ed.): *Adaptive Antenna Arrays: Trends and Applications*. Springer, Heidelberg (2004)
17. Pal, S., Das, S., Basak, A., Suganthan, P.N.: Synthesis of difference patterns for monopulse antennas with optimal combination of array-size and number of subarrays - A multi-objective optimization approach. *Progress in Electromagnetics Research, PIER B* 21, 257–280 (2010)
18. Zhou, A., Qu, B.-Y., Li, H., Zhao, S.-Z., Suganthan, P.N., Zhang, Q.: Multi-objective Evolutionary Algorithms: A Survey of the State-of-the-art. *Swarm and Evolutionary Computation* 1(1), 32–49 (2011)

# Supervised Machine Learning Approach for Bio-molecular Event Extraction\*

Asif Ekbal<sup>1</sup>, Amit Majumder<sup>2</sup>, Mohammad Hasanuzzaman<sup>3</sup>, and Sriparna Saha<sup>1</sup>

<sup>1</sup> Department of Computer Science and Engineering,  
Indian Institute of Technology Patna, India

{asif, sriparna}@iitp.ac.in

<sup>2</sup> Academy of Technology, Kolkata  
jobamit48@yahoo.co.in

<sup>3</sup> WBIDCL, Kolkata  
hasanuzzaman.im@gmail.com

**Abstract.** The main goal of biomedical text mining is to capture biomedical phenomena from textual data by extracting relevant entities, information and relations between biomedical entities such as proteins and genes. Most of the research in the related areas were focused on extracting only binary relations. In a recent past, the focus is shifted towards extracting more complex relations in the form of bio-molecular events that may include several entities or other relations. In this paper we propose a supervised approach that enables extraction, i.e. identification and classification of relatively complex bio-molecular events. We approach this as the supervised machine learning problems and use the well-known statistical algorithm, namely Conditional Random Field (CRF) that makes use of statistical and linguistic features that represent various morphological, syntactic and contextual information of the candidate bio-molecular trigger words. Firstly, we consider the problem of event identification and classification as a two-step process, first step of which deals with the event identification task and the second step classifies these identified events to one of the nine predefined classes. Thereafter, we perform event identification and classification together. Three-fold cross validation experiments on the Biomedical Natural Language Processing (BioNLP) 2009 shared task datasets yield the overall average recall, precision and F-measure values of 58.88%, 74.53% and 65.79%, respectively, for the event identification. We observed the overall classification accuracy of 59.34%. Evaluation results of the proposed approach when identification and classification are performed together showed the overall recall, precision and F-measure values of 59.92%, 54.25% and 56.94%, respectively.

## 1 Introduction

The past history of text mining (*TM*) shows the great success of different evaluation challenges based on carefully curated resources. All these challenges have significantly contributed to the progress of their respective fields. This has also been similar for bio-text mining (*bio-TM*). Some of the bio-text mining evaluation challenges include

---

\* All authors equally contributed for the paper.

the LLL [1], and BioCreative [2]. The first two shared tasks addressed the issues of bio-information retrieval (*bio-IR*) and bio-Named Entity Recognition (*bio-NER*), respectively. The JNLPBA and BioCreative evaluation campaigns were associated with the bio-information extraction (*bio-IE*). These two addressed the issues of seeking relations between bio-molecules. With the emergence of NER systems with performance capable of supporting practical applications, the recent interest of the bio-TM community is shifting toward IE.

Relations among biomedical entities (i.e. proteins and genes) are important in understanding biomedical phenomena and must be extracted automatically from a large number of published papers. Most researchers in the field of Biomedical Natural Language Processing (BioNLP) have focused on extracting binary relations, including protein-protein interactions (PPIs). These were addressed in the evaluation challenges like LLL and BioCreative. Binary relations are not sufficient for capturing biomedical phenomena in detail, and there is a growing need for capturing more detailed and complex relations. For this purpose, two large corpora, BioInfer and GENIA, were created.

Similarly to previous bio-text mining challenges (e.g., LLL and BioCreative), the BioNLP'09 Shared Task (also addressed bio-IE, but it tried to look one step further toward finer-grained IE. The difference in focus is motivated in part by different applications envisioned as being supported by the IE methods. For example, BioCreative aims to support curation of PPI databases such as MINT [3], for a long time one of the primary tasks of bioinformatics. The BioNLP'09 shared task contains simple events and complex events. Whereas the simple events consist of binary relations between proteins and their textual triggers, the complex events consist of multiple relations among proteins, events, and their textual triggers. Bindings can represent events among multiple proteins, and regulations can represent causality relations between proteins and events. These complex events are more informative than simple events, and this information is important in modeling biological systems, such as pathways. The primary goal of BioNLP-09 shared task [4] was aimed to support the development of more detailed and structured databases, e.g. pathway or Gene Ontology Annotation (GOA) databases, which are gaining increasing interest in bioinformatics research in response to recent advances in molecular biology.

In the present paper, we propose a supervised machine learning approach that enables the extraction of complex bio-molecular events from the medical texts. The main goal of event extraction is to detect the bio-molecular events from the texts and to classify them into nine predefined classes, namely *gene expression*, *transcription*, *protein catabolism*, *phosphorylation*, *localization*, *binding*, *regulation*, *positive regulation* and *negative regulation*. Evaluation with the BioNLP 2009 shared task datasets show the encouraging performance in all of our experimental settings.

## 2 Proposed Approach for Event Extraction

In this section we describe our proposed approach for event extraction that involves identification of complex bio-molecular events from the texts and classification of them

into some predefined categories of interest. We approach this problem from the supervised machine learning perspective, and use Conditional Random Field (CRF) that makes use of statistical and linguistic features that represent various morphological, syntactic and contextual information of the candidate bio-molecular trigger words. In our first attempt, we approach the problem of event extraction in two phases. In the first phase we identify the trigger words that designate events. In the second step, the identified event triggers are classified into predefined nine categories. In our second setting, we solve the problem of event extraction in one step by performing event identification and classification together. The training set is highly imbalanced. We filter out those sentences that don't contain any proteins. We use the same set of features (described in Section ) for both event trigger identification and classification<sup>1</sup>. This same set of features is also used when event extraction is performed in one step. We use the datasets which were tokenized, stemmed, Part-of-Speech (PoS) tagged and named entity (NE)-tagged, and provided in the CoNLL-X format<sup>2</sup>. We use the McClosky-Charniak parsed outputs<sup>3</sup> which were converted to the Stanford Typed Dependencies format.

We employ CRF [5] that is used to calculate the conditional probability of values on designated output nodes given values on other designated input nodes. The conditional probability of a state sequence  $s = \langle s_1, s_2, \dots, s_T \rangle$  given an observation sequence  $o = \langle o_1, o_2, \dots, o_T \rangle$  is calculated as:

$$P_{\Lambda}(s|o) = \frac{1}{Z_o} \exp\left(\sum_{t=1}^T \sum_{k=1}^K \lambda_k \times f_k(s_{t-1}, s_t, o, t)\right),$$

where,  $f_k(s_{t-1}, s_t, o, t)$  is a feature function whose weight  $\lambda_k$ , is to be learned via training. The values of the feature functions may range between  $-\infty, \dots, +\infty$ , but typically they are binary. To make all conditional probabilities sum up to 1, we must calculate the normalization factor,

$$Z_o = \sum_s \exp\left(\sum_{t=1}^T \sum_{k=1}^K \lambda_k \times f_k(s_{t-1}, s_t, o, t)\right),$$

which as in hidden Markov models (HMMs), can be obtained efficiently by dynamic programming.

We use the C++ based CRF++ package<sup>4</sup>, a simple, customizable, and open source implementation of CRF for segmenting /labeling sequential data.

<sup>1</sup> In our future work, we would like to investigate different feature sets for identification and classification.

<sup>2</sup> <http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/SharedTask/tools.shtml>

<sup>3</sup> <http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/SharedTask/tools.shtml>

<sup>4</sup> <http://crfpp.sourceforge.net>

## 2.1 Features for Event Extraction

We identify and use the following set of features for event extraction. All these features are automatically extracted from the training datasets without using any additional domain dependent resources and/or tools.

1. **Context words:** We use preceding and succeeding few words as the features. This feature is used with the observation that contextual information plays an important role in identification of event triggers.
2. **Root words:** Stems of the current and/or the surrounding token(s) are used as the features of the event extraction module. Stems of the words were provided with the evaluation datasets of training, development and test.
3. **Part-of-Speech (PoS) information:** PoS information of the current and/or the surrounding tokens(s) are effective for event trigger identification. PoS labels of the tokens were provided by the organizers with the datasets.
4. **Named Entity (NE) information:** NE information of the current and/or surrounding token(s) are used as the features. NE information was provided with the shared task datasets.
5. **Semantic feature:** This feature is semantically motivated and exploits global context information. This is based on the content words in the surrounding context. We consider all unigrams in contexts  $w_{i-3}^{i+3} = w_{i-3} \dots w_{i+3}$  of  $w_i$  (crossing sentence boundaries) for the entire training data. We convert tokens to lower case, remove stopwords, numbers, punctuation and special symbols. We define a feature vector of length 10 using the 10 most frequent content words. Given a classification instance, the feature corresponding to token  $t$  is set to 1 if and only if the context  $w_{i-3}^{i+3}$  of  $w_i$  contains  $t$ . Evaluation results show that this feature is very effective to improve the performance by a great margin.
6. **Dependency features:** A dependency parse tree captures the semantic predicate-argument dependencies among the words of a sentence. Dependency paths between protein pairs have successfully been used to identify protein interactions. In this work, we use the dependency paths to extract events. We use the McClosky-Charniak parses which are converted to the Stanford Typed Dependencies format and provided with the datasets. We define a number of features based on the dependency labels of the tokens.
  - **Dependency path from the nearest protein:** Dependency relations of the path from the nearest protein are used as the features. Let us consider a path from “phosphorylation” to “CD40” be “nsubj inhibits acomp binding prep to domain num”. Due to the large number of possible words, use of these words on the paths may lead to data sparsity problems, and in turn to poor generalization. Suppose we have a sentence with similar semantics, where the synonym word “prevents” is used instead of “inhibits”. If we use the words on the path to represent the path feature, we end up with two different paths for the two sentences that have similar semantics. Therefore, in this work we use only the dependency relation types among the words to represent the paths. For example, the path feature extracted for the (phosphorylation, CD40) trigger/participant pair is “nsubj acomp prep to num”.



- **Boolean valued features:** Two boolean-valued features are defined using the dependency path information. The first feature checks whether the current token's child is a proposition and the chunk of the child includes a protein. The second feature fires if and only if the current token's child is a protein and its dependency label is OBJ.
7. **Shortest path:** Distance of the nearest protein from the current token is used as the feature. This is an integer-valued feature that takes the value equal to the number of tokens between the current token and the nearest protein.
  8. **Word prefix and suffix:** Fixed length (say,  $n$ ) word suffixes and prefixes may be helpful to detect event triggers from the text. Actually, these are the fixed length character strings stripped either from the rightmost (for suffix) or from the leftmost (for prefix) positions of the words. If the length of the corresponding word is less than or equal to  $n-1$  then the feature values are not defined and denoted by ND. The feature value is also not defined (ND) if the token itself is a punctuation symbol or contains any special symbol or digit. This feature is included with the observation that event triggers share some common suffixes and/or prefixes. In this work, we consider the prefixes and suffixes of length up to four characters.
  9. **Named Entities in Context:** We calculate the frequencies of NEs within the various contexts of a sentence. This feature is defined with the observation that NEs appear most of the times near to the event triggers. Let us suppose that  $w$  is the current token and  $L$  is the size of the sentence in terms of the number of words. We consider various contexts as:  $\text{context-size} = L/K$ , where  $K$ : 1 to 5. Now, considering  $w$  as the centre we define a context window as:  $\text{context-window-size} = 2 * \text{context-size} + 1$ . When the size exceeds the length of the sentence, we added some empty slots and filled it by the class labels "Other-than-NEs" (denoted by O). For word  $w$ , a feature vector of length 5 is defined. Depending upon the value of  $K$ , the corresponding feature fires. The value is set equal to the number of NEs within the contexts of "context-window-size". For example, for  $K=1$ , the entire sentence is considered (i.e.,  $\text{context-size} = L$ ). For the first word of the sentence, the context window is equal to more than twice (i.e.,  $2 * \text{context-size} + 1$ ) of the sentence length. For  $K=2$ , the context-size is half of the sentence length. Again, centring the word  $w$  we define a context of double length by filling the preceding empty slots with O. The feature value is equal to the number of NEs within this window.

### 3 Datasets and Experimental Results

In this section we describe datasets used in our task and the experimental results.

#### 3.1 Datasets

We use the BioNLP-09 shared task datasets. The events were selected from the GENIA ontology based on their significance and the amount of annotated instances in the GENIA corpus. The selected event types all concern protein biology, implying that they

take proteins as their theme. The first three event types concern protein metabolism that actually represents protein production and breakdown. *Phosphorylation* represents protein modification event whereas *localization* and *binding* denote fundamental molecular events. *Regulation* and its sub-types, *positive* and *negative* regulations are representative of regulatory events and causal relations. The last five event types are universal but frequently occur on proteins. Detailed biological interpretations of the event types can be found in Gene Ontology (GO) and the GENIA ontology. From a computational point of view, the event types represent different levels of complexity.

Training and development datasets were derived from the publicly available event corpus [6]. The test set was obtained from an unpublished portion of the corpus. The shared task organizers made some changes to the original GENIA event corpus. Irrelevant annotations were removed, and some new types of annotation were added to make the event annotation more appropriate. The training, development and test datasets have 176,146, 33,937 and 57,367 tokens, respectively.

### 3.2 Experimental Results

We use CRF for training and testing. In order to properly denote the boundaries of events triggers we use standard BIO notation, B, I and O denote the beginning, internal and outside of tokens. For example, *interacting receptor-ligand pair* is annotated as *interacting/B-Event receptor-ligand/I-Event pair/I-Event* in the two-phase approach. The single word token is annotated with B-Event class. For example, *TRAF2 is a ... which binds/B-EVNET to the CD40 cytoplasmic domain*.

The system is tuned on the development data, and the results are reported using 3-fold cross validation<sup>5</sup>. The system is evaluated with the standard recall, precision and F-measure metrics. We followed the strict matching criterion, i.e. credit is given if and only if the event types are the same and the event triggers are the same.

A number of CRF models are generated by varying the available features and/or feature templates. Evaluation results on the development set showed the highest performance with the recall, precision and F-measure values of 59.30%, 46.66% and 52.23%, respectively for the event trigger identification task. Results suggest that apart from the context words, integrating other features within a larger context sometimes decreases the overall performance. In the second step, we classify the correctly detected events into the predefined nine event classes. For classification, we use the same set of features as that of event identification. Out of 847 events, 331 were found to be classified correctly, and thus producing an overall accuracy of 39.09%. We then perform identification and classification together, and their results show the overall recall, precision and F-measure values of 59.92%, 54.52% and 56.94%, respectively. We observe the fact that the same feature set doesn't perform well for all the tasks. Appropriate feature selection is, thus, essential for each task, i.e. identification, classification, as well as identification and classification both.

After finding the best configuration using the development set, we use this to perform 3-fold cross validation to report the final results. Initially, the training dataset is

<sup>5</sup> It is to be noted that due to the unavailability of gold-standard annotations we were unable to evaluate the system with the test dataset.

**Table 1.** Results of event detection for 3-fold cross validation (in %)

Fold number	recall	precision	F-measure
1	59.36	74.20	65.95
2	59.71	74.71	66.37
3	57.58	74.69	65.03
Average	58.88	74.53	65.79

**Table 2.** Results for event classification (in %)

Event type	recall	precision	F-measure
Gene_expression	86.03	50.69	63.79
Transcription	74.76	46.03	56.97
Protein_catabolism	90.47	62.67	74.05
Localization	81.03	57.75	67.44
Binding	75.36	41.95	53.89
Phosphorylation	98.12	69.79	81.56
Regulation	56.67	29.71	38.98
Positive_regulation	77.45	31.11	44.39
Negative_regulation	83.84	33.47	47.84
<b>Overall</b>	80.41	47.02	59.34

randomly splitted into nearly three equal subsets. Two subsets are used for training and the remaining one subset is withheld for testing. This process is repeated three times to perform 3-fold cross validation. Evaluation results of 3-fold cross validation for event detection are reported in Table 1 that shows the overall average recall, precision and F-measure values of 58.88%, 74.53% and 65.79%, respectively.

We then perform classification and report the evaluation results in Table 2. Evaluation results show the overall recall, precision and F-measure values of 80.41%, 47.02% and 59.34%, respectively. It shows, in general, high recall for all the event classes. But the relatively lower precisions are responsible for such an overall accuracy. The performance of the event classification phase directly depends on the event detection phase. Performance of event identification propagates through the pipeline and hurts the classification accuracy. Thus, the classification performance can be improved if the errors of detection can be reduced. Thereafter, we perform detection and classification together and report their 3-fold evaluation results in Table 3. Comparisons between Table 1, Table 2 and Table 3 clearly show that two-phase implementation (i.e., when detection and classification performed in a series) is better than one-pass implementation (i.e., when detection and classification performed together). Results also indicate that *gene expression*, *protein catabolism*, *localization*, *phosphorylation* and *transcription* are relatively easier for both identification and/or classification. In contrast, regulatory events, i.e. *regulation*, *positive regulation* and *negative regulation* are difficult to identify and/or classify. This led the importance of proper feature selection for event identification and classification both.

**Table 3.** Results for event detection and classification (in %)

Event type	recall	precision	F-measure
Gene_expression	77.12	52.64	62.58
Transcription	45.59	43.11	44.31
Protein_catabolism	63.51	62.68	63.09
Localization	52.10	58.83	55.26
Binding	53.04	42.49	47.18
Phosphorylation	79.67	68.34	73.57
Regulation	40.85	33.39	36.75
Positive_regulation	42.78	34.62	38.27
Negative_regulation	52.17	39.54	44.98
<b>Overall</b>	50.52	48.41	49.54

## 4 Conclusion

In this paper we have proposed a supervised machine learning approach for biological event extraction that involves identification of complex bio-molecular events and classification of them into the predefined nine classes. We have used CRF that exploits various statistical and linguistic features in the forms of morphological, syntactic and contextual information of the candidate bio-molecular trigger words. Firstly, we considered the problem of event detection and classification as a two-step process, first step of which dealt with the event detection whereas the second step was involved with the classification of these events. Thereafter, we treated event extraction problem as one-step process, and performed event detection and classification together. Three-fold cross validation experiments on the BioNLP 2009 shared task datasets yield the good performance in all our settings.

## References

1. Nedellec, C.: Learning Language in Logic -Genic Interaction Extraction Challenge. In: Cussens, J., Nedellec, C. (eds.) Proceedings of the 4th Learning Language in Logic Workshop (LLL 2005), pp. 31–37 (2005)
2. Hirschman, L., Krallinger, M., Valencia, A. (eds.): Proceedings of the Second BioCreative Challenge Evaluation Workshop. CNIO Centro Nacional de Investigaciones Oncologicas (2007)
3. Chatr-aryamontri, A., Ceol, A., Palazzi, L.M., Nardelli, G., Schneider, M.V., Castagnoli, L., Cesareni, G.: MINT: the Molecular INTeraction database. *Nucleic Acids Research* 35(suppl. 1), D572–D574 (2007)
4. Kim, J.-D., Ohta, T., Pyysalo, S., Kano, Y., Tsujii, J.: Overview of BioNLP 2009 shared task on event extraction. In: BioNLP 2009: Proceedings of the Workshop on BioNLP, pp. 1–9 (2009)
5. Pyysalo, S., Ginter, F., Heimonen, J., Bjorne, J., Boberg, J., Jarvinen, J., Salakoski, T.: BioInfer: A corpus for information extraction in the biomedical domain. *BMC Bioinformatics* 8, 50 (2007)
6. Kim, J.-D., Ohta, T., Tsujii, J.: Corpus annotation for mining biomedical events from literature. *BMC Bioinformatics* 9, 10 (2008)

# Design of Two Channel Quadrature Mirror Filter Bank: A Multi-Objective Approach

Subhrajit Roy<sup>1</sup>, Sk. Minhazul Islam<sup>1</sup>, Saurav Ghosh<sup>1</sup>, Shizheng Zhao<sup>2</sup>,  
Ponnuthurai Nagaratnam Suganthan<sup>2</sup>, and Swagatam Das<sup>1</sup>

<sup>1</sup>Dept. of Electronics and Telecommunication Engg.,  
Jadavpur University, Kolkata 700 032, India

<sup>2</sup>Dept. of Electronics and Electrical Engg.,  
Nanyang Technological University

{roy.subhrajit20, skminha.isl}@gmail.com,  
saurav\_online@yahoo.in, ZH0047NG@e.ntu.edu.sg,  
epnsugan@ntu.edu.sg, swagatamdas19@yahoo.co.in

**Abstract.** In Digital Signal processing domain the Quadrature Mirror Filter (QMF) design problem is one of the most important problems of current interest. While designing a Quadrature Mirror Filter the goal of the designer is to achieve minimum values of Mean Square Error in Pass Band (MSEP), Mean Square Error in Stop Band (MSES), Square error of the overall transfer function of the QMF bank at the quadrature frequency and Measure of Ripple (*mor*). In contrast to the existing optimization-based methods that attempt to minimize a weighted sum of the four objectives considered here, in this article we consider these as four distinct objectives that are to be optimized simultaneously in a multi-objective framework. To the best of our knowledge, this is the first time to apply MO approaches to solve this problem. We use one of the best known Multi-Objective Evolutionary Algorithms (MOEAs) of current interest called NSGA-II as the optimizer. The multiobjective optimization (MO) approach provides greater flexibility in design by producing a set of equivalent final solutions from which the designer can choose any solution as per requirements. Extensive simulations reported shows that results of NSGA-II is superior to that obtained by two state-of-the-art single objective optimization algorithms namely DE and PSO.

## 1 Introduction

The basic idea and the layout of QMF bank was first proposed by Johnston [1]. Since then research concerning a QMF bank has been carried out and it has found wide applications in various signal processing fields [2-5].

Recently researchers have proposed many different techniques for the accurate design of QMF banks [6-10]. Finding the accurate filter coefficients of the QMF banks is a complex problem and traditional analytical methods may fail to solve this problem. Thus, the use of derivative free evolutionary algorithms (EA) seems to be a powerful alternative for the traditional methods to solve the QMF problem. The most

popular EAs of current literature are Genetic Algorithms (GA) [11], Particle Swarm Optimization (PSO) [12], and Differential Evolution (DE) [13, 14]. These algorithms are suitable alternatives to the conventional methods because of their ability to deal with difficult problems featuring complex landscapes. These algorithms and their different variants find application in many real world problems [15, 16].

These single objective optimizers tackle all the objectives simultaneously by creating a single objective function by taking weighted sum of all of the objectives which may even be conflicting. In fact in [17] the QMF design problem has been tackled by the Particle Swarm Optimization technique where a weighted linear sum of all the design objectives was considered to form a single aggravated objective function. This weighted sum method is subjective and the solution obtained will depend on the values of the specified weights. It is difficult to find a universal set of weights that suits different instantiations of the same problem. So motivated by the inherent multi-objective nature of the QMF design problem and to remove the problems associated with using the weight factors we, to the best of our knowledge, for the first time use multi-objective algorithms [18] to solve this particular problem. Thus, we started to look for the most popular multi-objective algorithm that could solve this problem much more efficiently as compared to the conventional single-objective approaches. In this article we have employed the widely known NSGA-II algorithm [19] for the design of QMF banks.

In this work we employ NSGA-II to obtain the optimal coefficients of linear phase Quadrature Mirror Filter Bank while achieving the best possible design trade-offs between four objectives corresponding to minimum MSEP, minimum MSES, minimum Square error of the overall transfer function of the QMF bank at the quadrature frequency and minimum *mor*. The best tradeoff solutions, identified with a fuzzy membership based approach [20] over a certain instance of the design problem is shown to outperform the solutions achieved by single objective optimizers namely DE and PSO.

## 2 General Description of NSGA-II

NSGA-II [19] is a popular non-domination based genetic algorithm for multi-objective optimization which incorporates elitism and no sharing parameter needs to be chosen *a priori*. The population is initialized as usual. After the initialization of the population it is sorted based on non-domination into each front. The first front being completely non-dominant set in the current population and the second front being dominated by the individuals in the first front only and the front goes so on. Each individual of each of the front are assigned rank (fitness) values based on the front in which they belong to. Individuals in first front are assigned a fitness value of 1 and individuals in second are given fitness value as 2 and so on. In addition to fitness value a new parameter called *crowding distance* is calculated for each individual. The crowding distance is a measure of how close an individual is to its neighbours. Large average crowding distance will result in better diversity in the population. Parents are selected from the population by using binary tournament selection based on the rank and crowding distance. An individual is selected in the rank is lesser than the other or if crowding distance is greater than the other. The selected population generates

off-springs from crossover and mutation operators. The population with the current population and current off-springs is sorted again based on non-domination and only the best  $N$  individuals are selected, where  $N$  is the population size. The selection is based on rank and the on crowding distance on the last front.

### 3 Multi-Objective Formulation of the Design Problem

The basic block diagram of a typical two-channel QMF bank is shown in the Figure1.

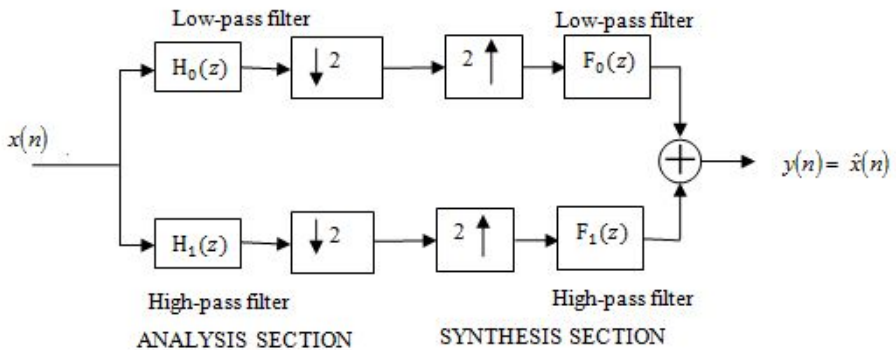


Fig. 1. Two Channel QMF Bank

This is basically a multi rate digital filter structure which splits the input signal  $x(n)$  into two sub-band signals having equal band width, using the low-pass and high-pass analysis filters  $H_o(z)$  and  $H_1(z)$ , respectively. These sub-band signals are down sampled by a factor of two to achieve signal compression or to reduce processing complexity. At the output side these two sub-band signals are interpolated by a factor of two and passed through low-pass and high-pass synthesis filters,  $F_o(z)$  and  $F_1(z)$ , respectively.

The  $z$ -transform of the output signal  $x(n)$  of the two-channel QMF bank can be written as [6]:

$$\hat{X}(z) = \frac{1}{2} [H_0(z)F_0(z) + H_1(z)F_1(z)]X(z) + \frac{1}{2} [H_0(-z)F_0(z) + H_1(-z)F_1(z)]X(-z) \quad (1)$$

The first term of the above equation represents a linear shift-invariant response between  $X(z)$  and  $\hat{X}(z)$  whereas the second term represents the aliasing error because of change in sampling rate.

Thus the aliasing effect is removed if the second term of Equation 1 becomes zero i.e. if

$$[H_0(-z)F_0(z) + H_1(-z)F_1(z)] = 0 \quad (2)$$

This condition is simply satisfied by setting  $F_0(z) = H_1(-z)$ ,  $-F_0(z) = F_1(z)$  and  $H_1(z) = H_0(-z)$ . The Equation 1 reduces to

$$\widehat{X}(z) = \frac{1}{2} [H_0(z) F_0(z) + H_1(z) F_1(z)] X(z) + \frac{1}{2} (0) X(-z) \tag{3.1}$$

or

$$\widehat{X}(z) = \frac{1}{2} [H_0^2(z) - H_0^2(-z)] X(z) \tag{3.2}$$

or

$$\widehat{X}(z) = T(z) X(z) \tag{3.3}$$

where,

$$T(z) = \frac{1}{2} [H_0^2(z) - H_0^2(-z)] \tag{4}$$

If we assume a linear-phase FIR low-pass filter with even length then the impulse response of the analysis section can be expressed as  $h_0(n) = h_0(N-n-1)$  where  $n = 0, 1, 2, \dots, (\frac{N}{2}-1)$ ,  $N$  is the length of the impulse response.

The frequency response of Equation 3 thus becomes

$$\widehat{X}(e^{j\omega}) = \frac{1}{2} e^{-j(N-1)\omega} [ |H_0(\omega)|^2 - (-1)^{N-1} |H_0(\pi - \omega)|^2 ] X(e^{j\omega}) \tag{5}$$

where

$$H_0(\omega) = \sum_{n=0}^{N/2-1} 2h_0(n) \cos(n - \frac{N-1}{2}) \tag{6}$$

The overall transfer function of QMF bank in frequency domain becomes

$$\frac{\widehat{X}(e^{j\omega})}{X(e^{j\omega})} = T(e^{j\omega}) = \frac{1}{2} [ |H_0(\omega)|^2 e^{-j\omega(N-1)} - |H_0(\omega - \pi)|^2 e^{-j(\omega-\pi)(N-1)} ] \tag{7}$$

or,  $T(e^{j\omega}) = \frac{1}{2} e^{-j\omega(N-1)} [ |H_0(\omega)|^2 - (-1)^{N-1} |H_0(\omega - \pi)|^2 ] \tag{8}$

or,  $T(e^{j\omega}) = \frac{1}{2} e^{-j\omega(N-1)} T'(\omega) \tag{9}$

where

$$T'(\omega) = [ |H_0(\omega)|^2 - (-1)^{N-1} |H_0(\omega - \pi)|^2 ] \tag{10}$$

This reveals that the QMF bank has a linear phase delay due to the term  $e^{-j\omega(N-1)}$  and the magnitude response  $T'(\omega)$  should be unity for the condition of perfect reconstruction.

Then the condition for perfect reconstruction is to minimize the four objective functions  $E_p$ ,  $E_s$ ,  $E_t$  and  $mor$  defined below:



$E_p$  is the mean square error in pass band (*MSEP*) which describes the energy of reconstruction error between 0 and  $\omega_p$ .

$$E_p = \frac{1}{\pi} \int_0^{\omega_p} |H_0(0) - H_0(\omega)|^2 d\omega \tag{11}$$

$E_s$  is the mean square error in stop band (*MSES*) which denotes the stop band energy related to LPF between  $\omega_s$  and  $\pi$  [6] as ,

$$E_s = \frac{1}{\pi} \int_{\omega_s}^{\pi} (|H_0(\omega)|^2) d\omega \tag{12}$$

$E_t$  is the square error of overall transfer function at quadrature frequency  $\frac{\pi}{2}$ .

$$E_t = \left[ H_0\left(\frac{\pi}{2}\right) - \frac{1}{\sqrt{2}} H_0(0) \right]^2 \tag{13}$$

Measure of ripple (*mor*) [21] is

$$\text{mor} = \max |10 \log_{10} |T'(\omega)| - \min |10 \log_{10} |T'(\omega)| \tag{14}$$

An MOEA will allow us to find the right balance between the four objectives shown above. When an MOEA is used then we get an approximation of the Pareto Front which contains numerous solutions. So an MOEA will allow us greater flexibility in designing a QMF bank because a single-objective EA gives us only one solution in one run which might not completely satisfy the designer's needs.

## 4 Experiments and Results

In this article, one instantiation of the design problem is solved in a MO framework by using the NSGA-II algorithm. We compare the best compromise solution obtained by NSGA-II with the best results achieved by the single-objective optimization techniques, namely DE and PSO. This DE variant is known as DE/rand/1/bin and is the most popular one in DE literature [13]. The PSO version used for comparison is the one used in the recently developed single objective QMF design problem employing PSO [17]. In what follows we report the best results obtained from a set of 50 independent runs of NSGA-II and its single-objective competitors, where each run for each algorithm is continued up to 10000 Function Evaluations (FEs). Note that for NSGA-II we extract the best compromise solution obtained with the fuzzy membership function based method outlined in [20]. For NSGA-II and the contestant algorithms we employ the best suited parametric set-up chosen with guidelines from their respective literatures.

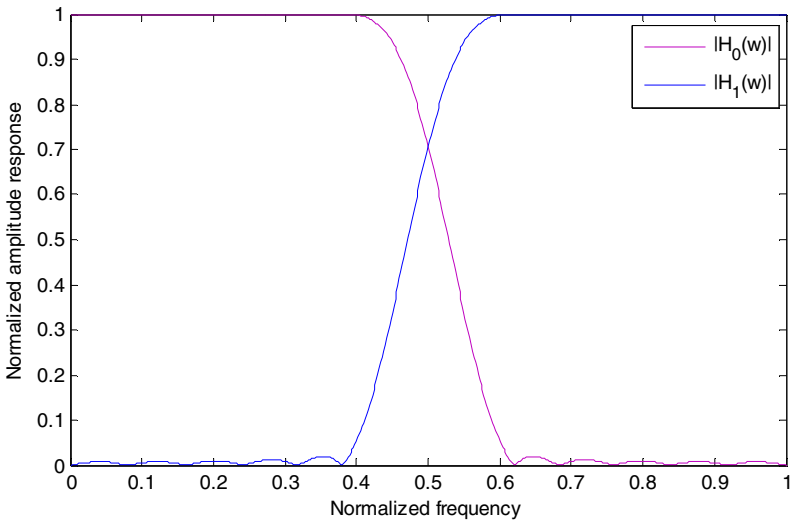
We consider a two-channel QMF bank for  $N = 24$ ,  $\omega_p = 0.4\pi$ ,  $\omega_s = 0.6\pi$  and with 12 filter coefficients. The performance of the considered algorithms are

evaluated in terms of  $E_p, E_s, E_t, mor$ , stop-band edge attenuation (*SBEA*) and stop-band first lobe attenuation (*SBFLA*).  $E_p, E_s, E_t$  and *mor* are explained in Section 3, whereas  $SBEA = -20 \log_{10}(H_0(\omega_s))$  as given in [6] and *SBFLA* is obtained from the respective attenuation characteristics. One thing is to be noted that we want lower values of  $E_p, E_s, E_t$  and *mor* and higher values of *SBEA* and *SBFLA*.

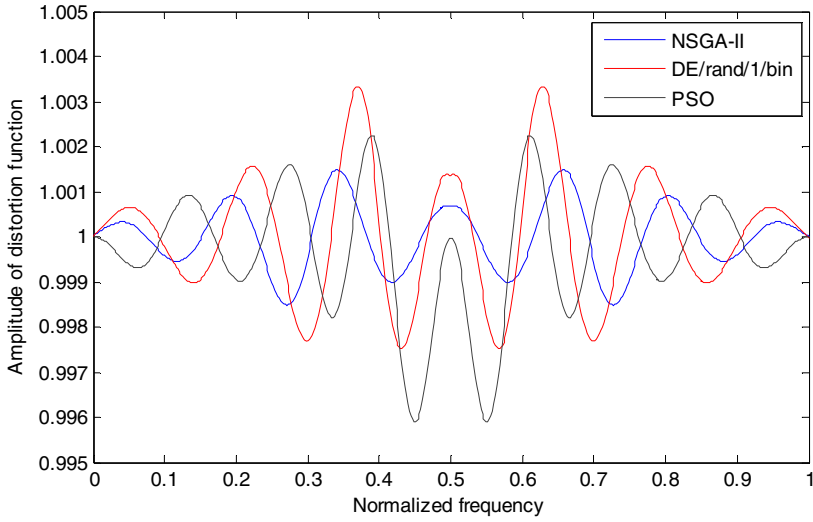
In Table 1 we provide the values of objectives  $E_p, E_s, E_t, mor$  and other two performance indices *SBEA* and *SBFLA* obtained by NSGA-II, DE/rand/1/bin and PSO. Table 1 clearly depicts that the best compromise solution obtained by NSGA-II is better than that obtained by the single objective optimizers DE/rand/1/bin and PSO thereby proving the superiority of our MO approach over the single-objective approach. Figure 2a portrays the normalized amplitude response for  $H_o, H_1$  filters of

**Table 1.** Design objectives, *SBEA* and *SBLFA* achieved with the three algorithms

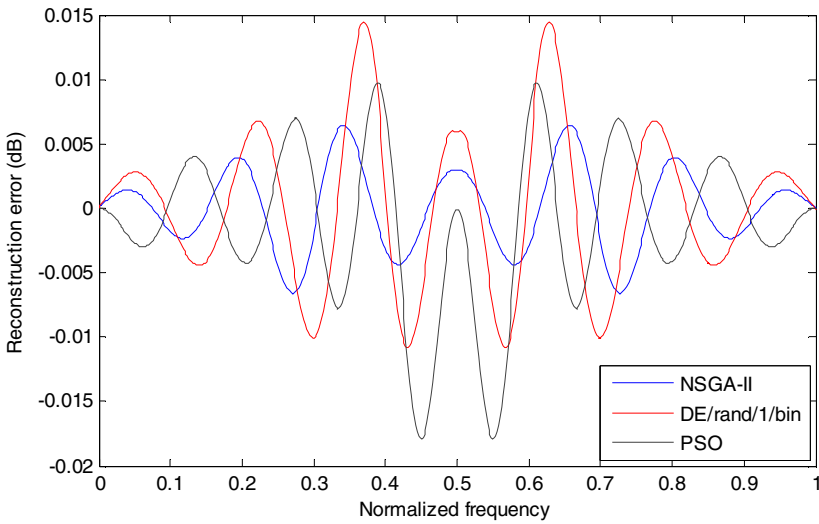
Algorithms	$E_p$	$E_s$	$E_t$	<i>mor</i>	<i>SBEA</i> (dB)	<i>SBLFA</i> (dB)
NSGA-II	<b>1.72e-08</b>	<b>3.06e-06</b>	<b>9.18e-07</b>	<b>1.19e-02</b>	<b>23.061</b>	<b>34.983</b>
DE/rand/1/bin	2.18e-08	3.82e-06	5.43e-06	1.91e-02	19.664	32.853
PSO	2.35e-08	5.79e-06	7.65e-06	1.48e-02	22.783	34.431



**Fig. 2a.** The normalized amplitude response for  $H_o, H_1$  filters of analysis bank



**Fig. 2b.** The attenuation characteristics of low pass filter  $H_o$



**Fig. 2c.** The reconstruction error of the QMF bank

analysis bank. The amplitude of distortion function  $T'(\omega)$  for all the considered algorithms is shown in Figure 2b. It is evident from the figure that the amplitude distortion is lowest for the filter obtained by NSGA-II. The reconstruction error in dB of the QMF banks are plotted in Figure 2c which clearly shows that the reconstruction error is least for the filter obtained by NSGA-II algorithm.

## 5 Conclusion

In this article we have proposed a new technique for designing the QMF bank problem as a multi-objective optimization framework and have employed the NSGA-II algorithm to solve the problem. The formulated MO problem has four design objectives: Mean Square Error in Pass band (MSEP), Mean Square Error in Stop band (MSES), Square error of the overall transfer function of the QMF bank at the quadrature frequency and Measure of Ripple (*mor*). One instantiation of the QMF design problem has been considered as example in this article. The results obtained by NSGA-II have been compared with two state-of-the-art single objective algorithms namely DE and PSO. The results obtained by the best compromise solution of NSGA-II have outperformed the results obtained by two state-of-the-art single objective optimization algorithms namely DE and PSO thereby proving the superiority of the MO framework of the QMF problem. Thus, unlike the single objective approaches the MO approach finally provide a set of design solutions that could allow the practitioner to satisfy the performance parameters. It gives a wide range of optimal solutions for the system under study. This MO framework is also robust and stable. This method is very effective and can be applied in practice to other filter design problems which are being treated till now as single objective problems. In our future work, we will investigate more design instances with other popular MO algorithms [22-24] as well as additional novel MO algorithms for solving this problem.

## References

1. Johnston, J.D.: A filter family designed for use in quadrature mirror filter banks. In: Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing, pp. 291–294 (1980)
2. Bellanger, M.G., Daguët, J.L.: TDM-FDM transmultiplexer: Digital Poly phase and FFT. IEEE Trans. Commun. 22(9), 1199–1204 (1974)
3. Gu, G., Badran, E.F.: Optimal design for channel equalization via the filter bank approach. IEEE Trans. Signal Process 52(2), 536–544 (2004)
4. Esteban, D., Galand, C.: Application of quadrature mirror filter to split band voice coding schemes. In: Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ASSP), pp. 191–195 (1977)
5. Liu, Q.G., Champagne, B., Ho, D.K.C.: Simple design of over sampled uniform DFT filter banks with application to sub-band acoustic echo cancellation. Signal Process 80(5), 831–847 (2000)
6. Chen, C.K., Lee, J.H.: Design of quadrature mirror filters with linear phase in the frequency domain. IEEE Trans. Circuits Syst. 39(9), 593–605 (1992)
7. Jou, Y.D.: Design of two-channel linear-phase quadrature mirror filter banks based on neural networks. Signal Process 87(5), 1031–1044 (2007)
8. Yu, Y.J., Lim, Y.C.: New natural selection process and chromosome encoding for the design of multiplier less lattice QMF using genetic algorithm. In: 8th IEEE International Conf. Electronics, Circuits and Systems, vol. 3, pp. 1273–1276 (2001)
9. Haddad, K.C., Stark, H., Galatsanos, N.P.: Design of two-channel equiripple FIR linear-phase quadrature mirror filters using the vector space projection method. IEEE Signal Process. Lett. 5(7), 167–170 (1998)

10. Bregovic, R., Saramaki, T.: A general purpose optimization approach for designing two-channel FIR filter banks. *IEEE Trans. Signal Process.* 51(7), 1783–1791 (2003)
11. Golberg, D.E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Massachusetts (1989)
12. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: *Proceedings of the IEEE International Conference Neural Networks*, vol. 4, pp. 1942–1948 (1995)
13. Storn, R., Price, K.V.: Differential Evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012, ICSI (1995), <http://http.icsi.berkeley.edu/~storn/litera.html>
14. Das, S., Suganthan, P.N.: Differential Evolution: A Survey of the State-of-the-Art. *IEEE Trans. Evolutionary Computation* 15(1), 4–31 (2011)
15. Zhao, S.Z., Willjuice, M.I., Baskar, S., Suganthan, P.N.: Multi-objective Robust PID Controller Tuning using Two Lbests Multi-objective Particle Swarm Optimization. *Information Sciences* 181(16), 3323–3335 (2011)
16. Pal, S., Das, S., Basak, A., Suganthan, P.N.: Synthesis of difference patterns for monopulse antennas with optimal combination of array-size and number of subarrays - A multiobjective optimization approach. *Progress in Electromagnetics Research, PIER B* 21, 257–280 (2010)
17. Upender, J.P., Gupta, C.P., Singh, G.K.: Design of two-channel quadrature mirror filter bank using particle swarm optimization. *Signal Processing* 20, 304–313 (2010), doi:10.1016/j.dsp.2009.06.014
18. Zhou, A., Qu, B.-Y., Li, H., Zhao, S.-Z., Suganthan, P.N., Zhang, Q.: Multi-objective Evolutionary Algorithms: A Survey of the State-of-the-art. *Swarm and Evolutionary Computation* 1(1), 32–49 (2011)
19. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6(2), 182–197 (2002)
20. Abido, M.A.: A novel multiobjective evolutionary algorithm for environmental/economic power dispatch. *Electric Power Systems Research* 65, 71–81 (2003)
21. Swaminathan, K., Vaidyanathan, P.P.: Theory and design of uniform DFT, parallel QMF banks. *IEEE Trans. Circuits Syst.* 33(12), 1170–1191 (1986)
22. Zhao, S.Z., Suganthan, P.N.: Two-lbests Based Multi-objective Particle Swarm Optimizer. *Engineering Optimization* 43(1), 1–17 (2011), doi:10.1080/03052151003686716
23. Qu, B.Y., Suganthan, P.N.: Multi-Objective Evolutionary Algorithms based on the Summation of Normalized Objectives and Diversified Selection. *Information Sciences* 180(17), 3170–3181 (2010)
24. Zhang, Q., Li, H.: MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Trans. Evolutionary Computation*, 712–731 (2007)

# Soft Computing Approach for Location Management Problem in Wireless Mobile Environment

Moumita Patra and Siba K. Udgata

Department of Computer and Information Sciences,  
University of Hyderabad, Hyderabad-500046, India  
patramoumita@yahoo.com, udgatacs@uohyd.ernet.in

**Abstract.** Location tracking and establishing end-to-end connectivity is one of the biggest challenges in mobile computing and wireless communication environment. Thus, there is a need to develop algorithms that can be easily implemented and used to solve a wide range of complex location management problems. Location management cost includes search cost and update cost. We have used reporting cells location management scheme to solve the location management problem. It has been reported that optimal reporting cell configuration is an *NP* complete problem. In the reporting cell location management scheme, few cells in the network are designated as reporting cells; mobile terminals update their positions (location update) upon entering one of these reporting cells. Vicinity of a reporting cell is defined as the number of reachable cells, without going through any other reporting cell. The objective of this paper is to minimize the location management cost of the network through an optimum reporting cell configuration. The proposed approach is giving better performance for bigger networks compared to earlier schemes. We also show the change in the location management cost with respect to different calls per mobility values and network size.

## 1 Introduction

Past few years have seen a tremendous growth in mobile computing, wireless communications and mobile services. Most of the research in mobile computing has addressed the issues involved in building and exploiting mobile and wireless systems, ranging from cellular, to satellite, to ad hoc and, more recently, to sensor networks. On the other hand, optimization research in mobile computing and wireless communications has received, in relative terms, less attention and was focussed mostly on optimizing the performance of individual communication protocols. Supporting mobility in all its aspects from cellular services to multimedia brings up new problems and opportunities for optimization research. One of the challenges of mobile computing research, is the tracking of the current location of users, which is the main component of location management. Location management involves paging, location updating and connection handover. The location update procedure enables the network to keep track of the current

location of the subscriber within the network, while paging is used to reach the hand-held device to which a call is destined. These two mechanisms guarantee that the mobile terminal can be reached even though there is no continuous active radio link between the mobile terminal and the network. The total cost of location management is the sum of location update cost and paging cost. The goal of location management is to find a strategy that balances the location update (registration) and paging (search) operation, so as to minimize the total cost of location tracking. One of the common location management strategies is the reporting cells strategy. In this strategy, a subset of cells in the network is designated as the reporting cells. Each mobile terminal performs a location update only when it enters one of these reporting cells. When a call arrives, the search is confined to the last updated reporting cell neighboring bounded non reporting cells. Optimal reporting cell planning is an NP- Complete problem [4]. In this paper, a new approach based on Genetic Algorithm(GA) and bounded vicinity value, is presented to obtain near optimal solution for the reporting cell planning problem.

## 2 Problem Formulation

Given a network, the objective is to minimize the total cost of location management. It is shown that optimal LA partitioning (one that gives the minimum location management cost) is an NP-complete problem [4]. Location management involves two operations, location update and location inquiry.

$$Totalcost = C \cdot N_{LU} + N_P \quad (1)$$

$C$  is a constant representing the cost ratio of location update and paging.  $C$  is said to have a value of 10, as it is considered that the cost of location update is 10 times more than that of location search [4],[5].  $N_{LU}$  is the total number of location updates performed during time  $T$  and  $N_P$  is the number of paging performed during time  $T$ . The following formula has been used for the total number of location updates and paging (performed during time  $T$ ):

$$N_{LU} = \sum_{i \in S} w_{mi} \quad (2)$$

$$N_P = \sum_{j=0}^{N-1} w_{cj} \cdot v(j) \quad (3)$$

Here  $w_{mi}$  denotes the movement weight associated with cell  $i$ ,  $w_{cj}$  represents the call arrival weight associated with cell  $j$  and  $v(j)$  is the vicinity of cell  $j$ . *Vicinity* is defined as the collection of all the cells that are reachable from a reporting cell  $i$  without entering another reporting cell. By substituting these values in the formula for total cost we get-

$$Totalcost = \sum_{i \in S} w_{mi} + \sum_{j=0}^{N-1} w_{cj} \cdot v(j)$$

In the above equation, the vicinity value  $v(j)$  calculation requires most of the computing time. Cost per call arrival is given as the total cost divided by the total number of call arrivals.

### 3 Literature Review

Riky et al, [1], have proposed the use of genetic algorithm(GA), tabu search(TS), and ant colony algorithm (ACA) to find the optimal or near optimal solutions to the reporting cells planning problem. Three cellular networks of 16, 36 and 64 cells are used to analyze the different algorithms (GA, TS and ACA). The cellular networks analyzed are small networks. In reality, the networks are much larger in size and number of cells.

Mehta et.al [2] have proposed the use of Simulated Annealing(SA) algorithm for solving the location management problem using the reporting cells planning scheme. Simulated Annealing is an optimization procedure based on the process of annealing, which is a process in which organized crystals are formed. During this process a physical substance is melted by raising to very high temperature, then cooled down slowly so that a long time is spent at each temperature drop which allows the molecules of substance to reach equilibrium state. The goal of the annealing process is to produce a stable minimal-energy final state. In this process, a physical substance usually moves from higher energy state to lower energy state. The SA algorithm is used for solving the reporting cells planning problem [9] and in this algorithm, energy(E) or objective function is analogous to location management cost. The algorithm has been applied on test networks with 19 and 36 cells. Simulation results show that simulated annealing algorithm can be effectively used to obtain near optimal results for reporting cell planning problem. It produced better results than two classical location management strategies - always-update and never-update. The main disadvantages in this method are similar to the ones given in the previous method i.e. the cellular networks analyzed are small networks and the vicinity value considered is much complex and takes the maximum computation time.

Karao glu et al [3] have proposed a zone-based scheme due to its wide usage in GSM systems. A comparison of three evolutionary algorithms namely genetic algorithm(GA), multi-population genetic algorithm (MPGA), and memetic algorithm(MA)[10] for location area management problem are presented. The objective is to assign cell to switch and cell to Location Area, so that the total cost is minimized. Two different networks with 256 base stations and with 576 base stations have been considered. Results show that when the algorithms are compared with respect to the unified cost value, memetic algorithm always gives the best results. When the execution times of the algorithms are considered, the memetic algorithm requires more computation due to the local search phase. Although memetic algorithm gives good results but it requires longer execution time in the search phase, which is not desirable.

Luz et.al[6] have proposed a new approach using Differential Evolution algorithm for the reporting cells planning problem, to minimize the total location



management cost. Twelve distinct test networks have been used and the work tries to find the best values of the differential evolution parameters and respective schemes. This algorithm has a key strategy to generate new individuals by calculating vector differences between other randomly-selected individuals of the population. The results obtained are better than the ones obtained using classical location management strategies as always-update and never-update. It gives better or similar results when applied on realistic networks. Xie et. al. [5] and Taheri et. al [11] also proposed dynamic location management strategy where the reporting cell configuration keeps changing with respect to time and change in traffic pattern.

## 4 Proposed Bounded Vicinity Method

Finding the vicinity of a reporting cell is considered to be a computationally intensive task. For a large network, the vicinity of a reporting cell can be restricted by considering a maximum bound on the upper limit. The algorithm of the proposed method based on Genetic Algorithm [7][8][9] is shown in algorithm-1:

---

### Algorithm 1. Proposed bounded vicinity approach

---

Initialize population  $P$  with random reporting cells

Initialize vicinity upper bound  $k$

```

for each reporting cell in the network do
  calculate vicinity  $v$ 
  while  $v > k$  do
    Choose a non reporting cell  $x$  from the neighboring cells
    Make  $x$  a reporting cell
    Calculate new vicinity  $n$ 
    let  $v=n$ 
  end while
end for
while stopping conditions not true do
  Evaluate population  $P$  using GA
end while

```

---

A population of 100 chromosomes is taken and each chromosome consists of a binary string representing a reporting cell configuration. This initial population is evaluated and after this it enters the evolution loop to find the further generations until the stopping conditions are met. We have used the proposed approach for different network sizes with different network traffics and have also compared it with the existing data set given in [1]. The different data sets for different networks have been generated randomly.

## 5 Results and Discussions

### 5.1 Simulations

The simulations have been carried out in a computer having Intel(R) Core(TM)2 Duo CPU and 3.00 GHz processor. The necessary code is developed using Matlab 7.10.0.

### 5.2 Results

In this strategy, we have shown that by considering a bounded constraint to the vicinity of a reporting cell, the total location management cost of the network decreases. We have taken the data set given in [1] and a random data set, and

**Table 1.** Results for a 4x4 network

Existing Data Set				
Method	Average	Minimum	Maximum	Deviation
Existing	12.253	12.252	12.373	0.986
Proposed	12.255	12.252	12.273	0.008
Random Data Set				
Method	Average	Minimum	Maximum	Deviation
Existing	7.516	7.294	8.651	0.600
Proposed	7.214	7.211	7.294	0.002

**Table 2.** Results for a 6x6 network

Existing Data Set				
Method	Average	Minimum	Maximum	Deviation
Existing	11.511	11.471	12.030	4.867
Proposed	11.474	11.471	11.573	0.014
Random Data Set				
Method	Average	Minimum	Maximum	Deviation
Existing	9.522	9.378	9.656	0.155
Proposed	9.442	9.378	9.653	0.008

**Table 3.** Results for a 8x8 network

Existing Data Set				
Method	Average	Minimum	Maximum	Deviation
Existing	14.005	13.782	14.671	0.600
Proposed	13.809	13.782	14.042	0.045
Random Data Set				
Method	Average	Minimum	Maximum	Deviation
Existing	11.516	11.420	11.595	0.064
Proposed	11.383	11.377	11.458	0.009

have compared the existing method [1] with the proposed method. We have given a limit to vicinity value from  $k=5$  to  $k=10$ , and the best results using the proposed approach are given in the Table-1, Table-2 and Table-3.

Table.1 shows the results for a 4x4 network. It can be seen that the proposed approach gives almost similar results when the existing data set is used as in [1], but the deviation is much less, and it gives much lesser location management cost than the existing approach, using a random data set. Similar results are obtained in case of 6x6 and 8x8 networks, as can be seen from tables 2 and 3, respectively. In reality, the network size are much bigger than the ones with which experiments

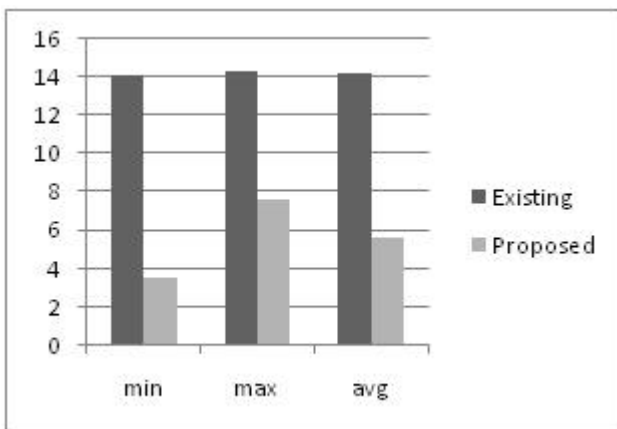


Fig. 1. Variation of cost for 10x10 Network

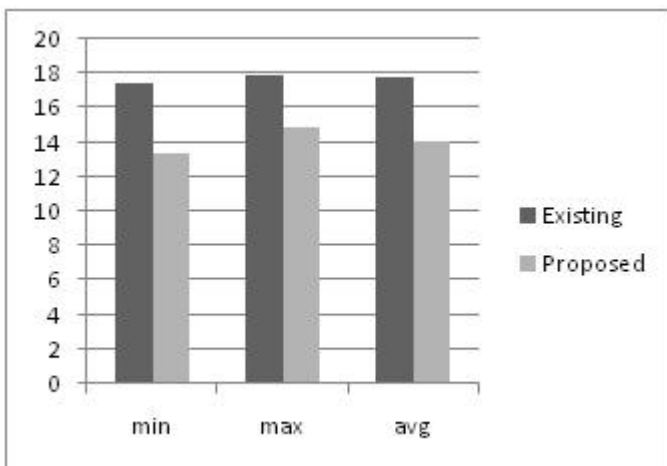


Fig. 2. Variation of cost for 20x20 Network

were carried out in the literature. So far, not much work has been done on bigger networks. We have used the existing as well as proposed methods on bigger networks consisting of 100 and 400 cells with random network traffic. The results obtained are shown in the following figure-1 and figure-2.

Fig.1 and 2, depict the results obtained by using the existing and the proposed approach for a 10x10 and 20x20 network, respectively. The graphs show that the location management cost decreases significantly by using the proposed approach as the network size grows.

### 5.3 Calls per Mobility

Calls per mobility is also an important parameter in wireless mobile computing environment. It is defined as the ratio of the total number of call arrivals in a network to the total number of user mobility in the network. It can be shown that the location management cost varies according to the change in calls per mobility. We have compared the location management costs obtained using different calls per mobility value, by taking different  $w_{mi}$  and  $w_{ci}$  values, using the existing as well as proposed approach. The results for a 8x8 network are shown in Figure-3. By considering different values of calls per mobility (cpm) of a 8x8 network, using existing as well as proposed method, we observe that after a certain value of cpm, there is not much change in the location management cost obtained. We have experimented with different cpm values for 4x4, 6x6 and 8x8 networks compared the obtained location management costs using existing approach and proposed approach. A similar trend is observed for all network sizes. We have only shown the trend for a 64 (8 X 8) cell network in the Figure-3.

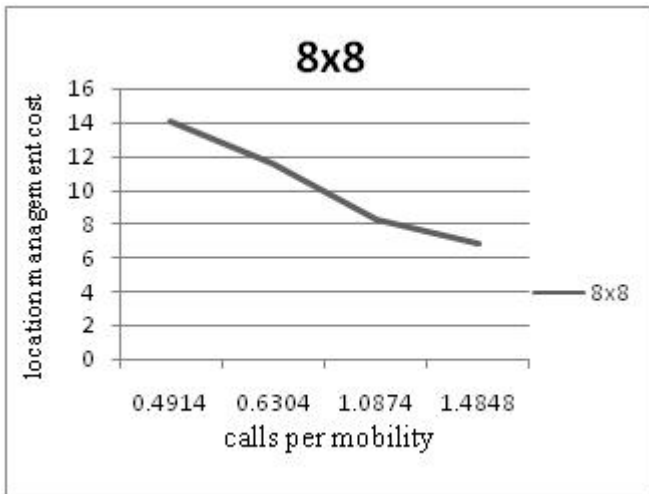


Fig. 3. Variation of cost with cpm for 8x8 network

Figure 3 shows that the location management cost decreases as the calls per mobility value increases and Table 4 gives a comparison of the location management cost for varying cpm values for an 8x8 network, using both the existing as well as proposed method. The results depict that the proposed method performs better than the existing approach, when the cpm value is less and it gives almost similar values when the cpm value increases.

**Table 4.** Variation of cost with cpm for 8x8 Network

cpm	Existing	Proposed
0.491	14.005	13.809
0.630	11.516	11.383
1.087	8.289	8.285
1.495	6.866	6.824

## 6 Conclusions

The proposed methodology tries to optimize the location management cost and performs better as compared to the methods given in the literatures. In comparison to the existing best reported results, our proposed bounded vicinity value approach gives exactly the same location management cost when applied to smaller networks of size of 16 (4x4), 36 (6x6) and 64 (8x8) cells. Unlike the earlier papers, we experimented with larger networks with 100 and 400 cells and the proposed method is able to optimize the location management cost. When used on different network sizes with different communication traffic, proposed method performs better compared to the existing methods. The calls per mobility value also plays an important role in the location management cost. We observed that with the increase in the calls per mobility value, the location management cost decreases up to a particular level and then becomes stable. Our proposed approach also performs better for lower calls per mobility value (i.e for a highly mobile network where the users are changing their locations frequently) in comparison to the existing approach.

## References

- [1] Subrata, R., Zomaya, A.Y.: A Comparison of Three Artificial Life Techniques for Reporting Cell Planning in Mobile Computing. *IEEE Transactions on Parallel and Distributed Systems* 14(2), 142–153 (2003)
- [2] Mehta, F., Swadas, P.: A Simulated Annealing Approach to Reporting Cell Planning Problem of Mobile Location Management. *International Journal of Recent Trends in Engineering* 2(2), 98–102 (2009)
- [3] Karaoğlu, B., Topçuoğlu, H., Gürgen, F.: Evolutionary Algorithms for Location Area Management. In: Rothlauf, F., Branke, J., Cagnoni, S., Corne, D.W., Drechsler, R., Jin, Y., Machado, P., Marchiori, E., Romero, J., Smith, G.D., Squillero, G. (eds.) *EvoWorkshops 2005. LNCS*, vol. 3449, pp. 175–184. Springer, Heidelberg (2005)

- [4] Gondim, P.R.L.: Genetic Algorithms and the Location Area Partitioning Problem in Cellular Networks. In: Proc. IEEE 46th Vehicular Technology Conf., pp. 1835–1838 (1996)
- [5] Xie, H., Tabbane, S., Goodman, D.J.: Dynamic Location Area Management and Performance Analysis. In: Proc. 43rd IEEE Vehicular Technology Conf. Personal Comm. Freedom Through Wireless Technology, pp. 536–539 (1993)
- [6] Almeida-Luz, S.M., Vega-Rodriguez, M.A., Gmez-Plido, J.A., Snchez-Prez, J.M.: Differential evolution for solving the mobile location management. *Applied Soft Computing* 11, 410–427 (2011)
- [7] Golberg, D.E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading (1989)
- [8] Michalewicz, Z.: *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, Berlin (1994)
- [9] Mitchell, M.: *An Introduction to Genetic Algorithms*. MIT Press, Cambridge (1996)
- [10] Moscato, P.A.: On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms, Tech. Rep. Caltech Concurrent Computation Program Report 826, Caltech (1989)
- [11] Taheri, J., Zomaya, A.Y.: Clustering techniques for dynamic mobility management. In: *MobiWac 2006: Proceedings of the 4th ACM International Workshop on Mobility Management and Wireless Access*, pp. 10–17. ACM (2006)

# Distribution Systems Reconfiguration Using the Hyper-Cube Ant Colony Optimization Algorithm

A.Y. Abdelaziz<sup>1</sup>, Reham A. Osama<sup>1</sup>, S.M. El-Khodary<sup>1</sup>, and Bijaya Ketan Panigrahi<sup>2</sup>

<sup>1</sup> Department of Electrical Power & Machines, Faculty of Engineering,  
Ain Shams University, Cairo, Egypt

<sup>2</sup> Department of Electrical Engineering, Indian Institute of Technology, Delhi, India

**Abstract.** This paper introduces the Ant Colony Optimization algorithm (ACO) implemented in the Hyper-Cube (HC) framework to solve the distribution network minimum loss reconfiguration problem. The ACO is a relatively new and powerful intelligence evolution method inspired from natural behavior of real ant colonies for solving optimization problems. In contrast to the usual ways of implementing ACO algorithms, the HC framework limits the pheromone values by introducing changes in the pheromone updating rules resulting in a more robust and easier to implement version of the ACO procedure. The optimization problem is formulated taking into account the operational constraints of the distribution systems. Results of numerical tests carried out on two test systems from literature are presented to show the effectiveness of the proposed approach.

## 1 Introduction

Distribution Network reconfiguration is the process of changing the topology of distribution systems by altering the open/closed status of switches to transfer loads among the feeders. Two types of switches are used in primary distribution systems. There are normally closed switches (sectionalizing switches) and normally opened switches (tie switches). Those two types of switches are designed for both protection and configuration management. In 1975, the network reconfiguration for loss reduction concept was first introduced by A. *Merlin and H.Back* [1] by applying the branch and bound heuristic technique. Later several reconfiguration techniques have been proposed which can be grouped into 3 main categories: those based on mathematical optimization techniques, those based upon purely heuristics and finally, techniques based on Artificial Intelligence and modern heuristics such as: Genetic Algorithms [2], particle swarm optimization [3], Simulated Annealing [4], Tabu Search [5, 6], etc. A survey of the early state of art is provided in [7]. In this paper, a method employing ACO implemented in the new HC framework is adopted to solve the network reconfiguration problem. The ACO is a population based approach that was proposed by *M.Dorigio* [8]. It is inspired by the foraging behavior of real ant colonies finding the shortest path from food sources to the nest without using visual cues. The paper is organized as follows; Section (2) explains the distribution system minimum loss reconfiguration problem, its objective function and constraints; Section

(3) illustrates the ACO paradigm; Section (4) introduces the HC-ACO, how it differs from the standard ACO and how it is applied to the reconfiguration problem; Section (5) shows the numerical results of applying the HC-ACO to two test systems from literature. Finally the conclusion is given in Section (6).

## 2 Formulation of the Network Reconfiguration Problem for Loss Reduction

The reconfiguration problem can be formulated as follows:

$$\text{Min } f = \sum_{i=1}^{N_R} R_i |I_i|^2 \quad (1)$$

Subject to the following constraints:

1. The voltage magnitude

$$V_{\min} \leq |V_i| \leq V_{\max} \quad ; \forall i \in N_b \quad (2)$$

2. The current limit of branches

$$|I_i| \leq I_{j\max} \quad ; \forall j \in N_R \quad (3)$$

3. Radial Topology

where  $f$  is the fitness function to be minimized corresponds to the total power loss in the system,  $R_i$  is the resistance of the branch  $i$  and  $I_i$  is the magnitude of the current flowing through the branch  $i$ ,  $V_i$  is the voltage on bus  $i$ ,  $V_{\min}$  and  $V_{\max}$  are minimum and maximum bus voltage limits respectively,  $I_i$  and  $I_{j\max}$  are current magnitude and maximum current limit of branch  $i$  respectively and  $N_b$  and  $N_R$  are the total number of buses and branches in the system respectively. The objective Function is calculated starting from the solution of the power flow equations that can be solved using the Forward/Backward Sweep method [9]. This method has excellent convergence characteristics and is very robust and proved to be efficient for solving radial distribution networks.

## 3 Ant Colony Optimization

### 3.1 Behavior of Real Ants

Initially, ants wander randomly and upon finding food return to their colony while laying down pheromone trails. If other ants find such a path, they are likely not to keep travelling at random, but to instead follow the trail, returning and reinforcing it if they eventually find food. Over time, however, the pheromone trail starts to evaporate, thus reducing its attractive strength. The more time it takes for an ant to travel down the path and back again, the more time the pheromones have to evaporate. A short path, by comparison, gets marched over faster, and thus the pheromone density remains high as it is laid on the path as fast as it can evaporate.



Each ant probabilistically prefers to follow a direction rich in pheromone rather than a poorer one. The indirect communication between the ants via the pheromone trails allows them to find the shortest paths between their nest and food sources and this behavior forms the fundamental paradigm of the ant colony search algorithm.

### 3.2 ACO Paradigm

In the ACO method, a set of artificial ants cooperate in finding optimal solutions to difficult discrete optimization problems. These problems are represented as a set of points called “states” and the ants move through adjacent states. Exact definitions of state and adjacency are problem specific. The ACO adopts three main rules:

#### 1. The State Transition Rule (“Random Proportional Rule”)

At first, each ant is placed on a starting state. Each will build a full path from the beginning to the end state through the repetitive application of the state transition rule given in (4)

$$P_k(i,j) = \frac{[\tau(i,j)]^\alpha [\eta(i,j)]^\beta}{\sum_{m \in J_k(i)} [\tau(i,m)]^\alpha [\eta(i,m)]^\beta}, \forall j \in J_k(i) \quad (4)$$

where  $P_k(i,j)$  is the probability with which ant  $k$  in node  $i$  chooses to move to node  $j$ ,  $\tau(i,j)$  is the pheromone which deposited on the edge between nodes  $i$  and  $j$ ,  $\eta(i,j)$  is the visibility of the edge connecting nodes  $i$  and  $j$  which is problem specific (e.g. inverse of the edge distance),  $J_k(i)$  is the set of nodes that remain to be visited by ant  $k$  positioned on node  $i$ .  $\alpha$  and  $\beta$  are parameters that determine the relative importance of pheromone versus the path’s visibility. The state transition rule favors transitions toward nodes connected by shorter edges with greater amount of pheromone.

#### 2. Local Updating Rule

While constructing the solution, each ant modifies the pheromone on the visited path. It is an optional step intended to shuffle the search process. It increases the exploration of other solutions by making the visited lines less attractive

$$\tau(i,j) = (1 - \rho) \tau(i,j) + \rho \tau_0 \quad (5)$$

where  $\tau(i,j)$  is the amount of pheromone deposited on the path connecting nodes  $i$  and  $j$ ,  $\tau_0$  is the initial pheromone value and  $\rho$  is a heuristically defined parameter.

#### 3. Global Updating Rule

When all tours are completed, the global updating rule is applied to edges belonging to the best ant tour providing a greater amount of pheromone to shortest tour.

$$\tau(i,j) = (1 - \sigma) \tau(i,j) + \sigma \delta^{-1} \quad (6)$$

where  $\delta$  is a parameter belonging to the globally best tour and  $\sigma$  is the pheromone evaporation factor element in the interval [0 1]. This rule is intended to make the search more directed enhancing the capability of finding the optimal solution.

## 4 Formulation of ACO in the Hyper-Cube (HC) Framework for Solving Minimum Loss Reconfiguration Problem

The HC framework is a recently developed framework for the standard ACO [10, 11]. It is based on changing the pheromone update rules used in ACO algorithms so that the range of pheromone variation is limited to the interval [0-1], thus providing automatic scaling of the auxiliary fitness function used in the search process and resulting in a more robust and easier to implement version of the ACO procedure. The distribution system is represented as an undirected graph  $G(B, L)$  composed of set  $B$  of nodes and a set  $L$  of arcs indicating the buses and their connecting branches (switches). Artificial ants move through adjacent buses, selecting switches that remain closed to minimize the system power losses [12, 13]. The solution iterates over three steps:

### 1. Initialization

The Solution starts with encoding parameters by defining:

- **System Parameters:** set of supply substations  $S$ , set of buses  $N_B$ , set of branches  $N_R$ , (where each branch has 2 possible states either “0” for an opened tie switch or “1” for a closed sectionalizing switch), load data  $P_{load}$ ,  $Q_{load}$ , branch data  $R_b$ ,  $X_b$ , base configuration of the system  $C^{(0)}$  which is defined by the system's tie switches, initial power losses of the system  $f(C^{(0)})$  by solving the power flow for  $C^{(0)}$  and evaluating the fitness function  $f$ .
- **Algorithm parameters:** Number of artificial ants in each iteration  $N$ , initial pheromone quantity  $\tau_0$  assigned to each switch, evaporation factor of pheromone trails  $\rho$ , the parameters  $\alpha$  and  $\beta$  that determine the relative importance of the line's pheromone versus its visibility, a counter  $h$  for the number of iterations, a counter  $x$  that is updated at the end of the iteration with no improvement in the objective function, maximum number of iterations  $H_{max}$ , and maximum number of iterations  $X_{max}$  with no improvement in the objective function respectively.

The base configuration is then set as an initial reference configuration and as the best configuration found so far such that  $C_{best} = C_{best}^{(0)} = C^{(0)}$ .

### 2. Ants' reconfiguration and pheromone updating

In each iteration  $h$ , a reference configuration is set as the best configuration of the previous iteration such that  $C_{best}^{(h-1)} = C_{ref}^{(h)}$ .  $N$  Ants are initially located on  $N$  randomly chosen open switches and are sent in parallel in such a way that each ant  $n$  in the  $h^{\text{th}}$  iteration introduces a new radial configuration  $C_n^{(h)}$  by applying the state transition rule. Once all ants finish their tour, the configuration corresponding to each ant is evaluated by computing the objective function  $f(C_n^{(h)})$ . The best configuration of the  $h^{\text{th}}$  iteration  $C_{best}^{(h)}$  is identified which is the configuration corresponds to the minimum evaluated objective function of all ants (minimum power loss). The best configuration of the  $h^{\text{th}}$  iteration  $C_{best}^{(h)}$  is compared to the best configuration so far  $C_{best}$  such that if  $f(C_{best}^{(h)}) < f(C_{best})$ , the overall best configuration is updated such that  $C_{best} = C_{best}^{(h)}$ . Finally, the pheromone updating rules are applied such that for all switches that belong to the best configuration, the pheromone values are updated using (7). Otherwise, the pheromone is updated using (8).

$$\tau^{(h)} = (1 - \rho) \tau^{(h-1)} + \rho \sigma \quad (7)$$

$$\tau^{(h)} = (1 - \rho) \tau^{(h-1)} \quad (8)$$

where  $\tau^{(h)}$  is the new pheromone value after the  $h^{\text{th}}$  iteration,  $\tau^{(h-1)}$  is the old value of pheromone after the  $(h^{\text{th}} - 1)$  iteration,  $\rho$  is arbitrarily chosen from the interval [0-1] and  $\sigma$  is a heuristically defined parameter which was chosen to be equal  $(f(C_{best}) / f(C_{best}^{(h)}))$  since  $f(C_{best}^{(h)})$  cannot be lower than  $f(C_{best})$  the pheromone assigned to any switch cannot fall outside the range [0-1] so that the pheromone update mechanism is fully consistent with the requirements of the HC framework [14].

### 3. Termination of the algorithm

The solution process continues until maximum number of iterations is reached  $h=H_{max}$ , or until no improvement of the objective function has been detected after specified number of iterations  $x=X_{max}$ .

---

#### *Pseudo code for the iterative process of the HC-ACO*

---

*Initialize parameters*

*While (Termination criteria is not met)*

*Update the reference configuration*

*While (i < Number of ants)*

*Ants construct the solution without violation of constraints*

*Fitness function is evaluated for each new configuration*

*End*

*Determine the best ant and update best configuration.*

*Apply pheromone updating rules.*

*End*

*Output the best configuration.*

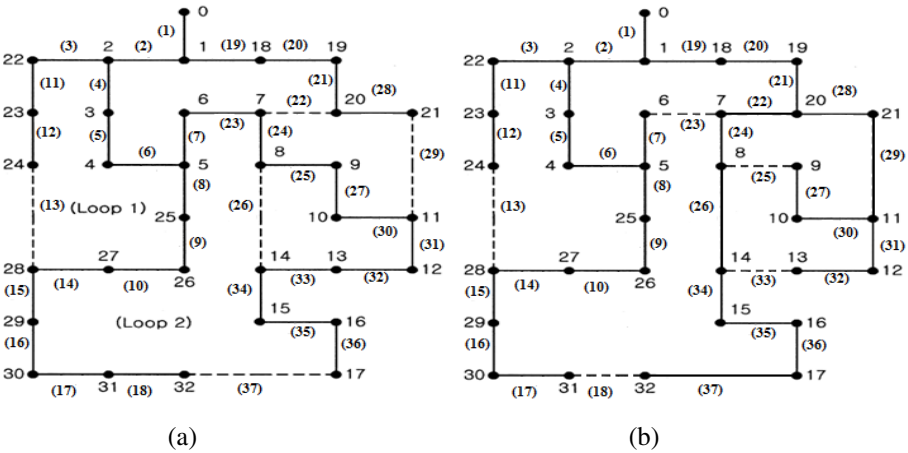
---

## 5 Worked Examples

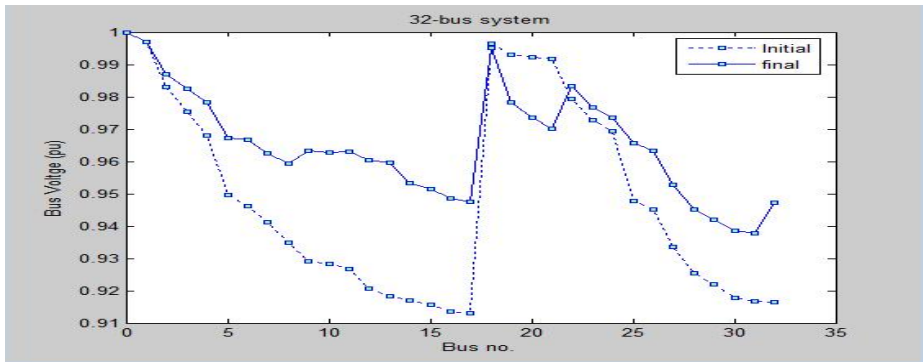
Two test systems from literature are investigated using the HC-ACO algorithm and the results are compared with previously applied algorithms. The solution algorithm was implemented using MATLAB V7. The first system is a 12.66 KV radial distribution system whose data are given in [15]. The system has one supply point, 32 buses, 3 laterals and 5 tie switches. The total substation loads of the base configuration are 3715 KW and 2300 KVAR. The base configuration of the system which is shown in Fig. (1a) has real power loss of 203 KW is [7-20, 8-14, 11-21, 17-32, 24-28] defined by the system's tie switches. The HC-ACO parameters used are  $N=10$ ,  $\alpha=0.1$ ,  $\beta=0.9$ ,  $\rho=0.04$ ,  $\tau_0=1$ ,  $H_{max}=100$  and  $W_{max}=10$ . The optimal configuration obtained by the proposed algorithm [6-7, 13-14, 8-9, 31-32, 24-28] which is shown in Fig. (1b) has a real power loss of 139.8 KW. This amounts to a reduction of 31.13 % in total power loss. Fig. (2) shows the voltage profile of the initial and final configurations of the system. As shown, the system's voltage profile is improved after reconfiguration such that before reconfiguration the lowest bus voltage was 0.9129 PU while after reconfiguration the lowest bus voltage is 0.9378 PU with 2.6% improvement.

**Table 1.** Evolution of the objective function, maximum and minimum pheromone values during the iterative process

Iteration no	1	2	3	4	5
Power losses (KW)	149.1	142.6	139.8	139.5	139.5
Max pheromone	1	1	1	1	1
Min.pheromone	0.96	0.921	0.8847	0.849	0.8153



**Fig. 1.** Schematic diagram of the 32-bus system (a) Initial configuration (b) final configuration. Tie switches and sectionalizing switches are represented by dotted and solid lines respectively.



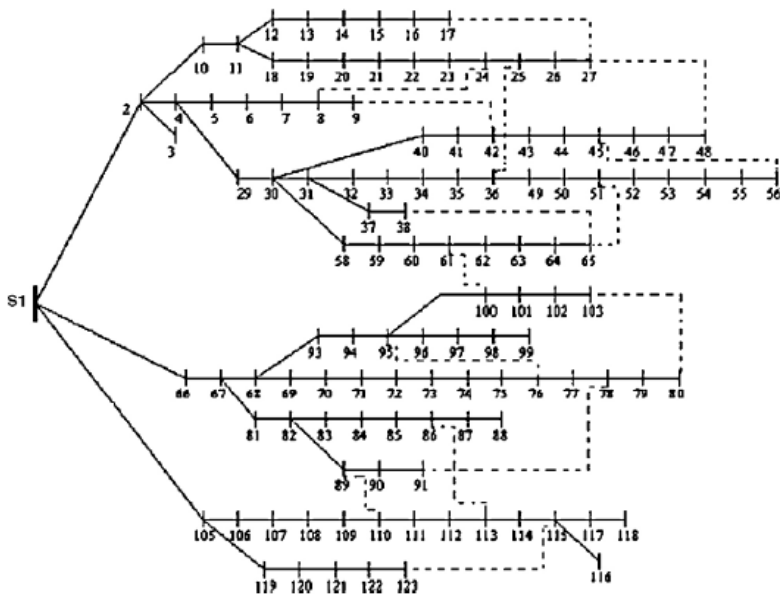
**Fig. 2.** Voltage profile of the 32-bus system before and after reconfiguration

The initial pheromone level is fixed  $\tau_0 = 1$  so that any switch belonging to all the best configurations found in any iteration its pheromone will always remain unity in all iterations. Conversely, if a switch does not belong to the best configuration for at

least one iteration, its pheromone level at the end of the iterative process will be lower than the initial value. Table (1) shows the evolution of the objective function, maximum and minimum pheromone values during the iterative process. As shown in Table (1), the optimum configuration was reached at the 4<sup>th</sup> iteration. The maximum pheromone value remains unity due to the fact that the best configuration has been at least reached or improved at every iteration and that is why implementing the ACO in the HC framework made the pheromone trails easy to handle.

**Table 2.** Results of reconfiguration of the 32-bus system

Reconfiguration	Losses(KW)	Tie Switches
Initial	203	7-20, 8-14, 11-21, 17-32, 24-28
Final using HC-ACO	139.5	6-7, 13-14, 8-9, 31-32, 24-28
Final using MPS[3], SA+TS [4], MTS[5]	139.5	6-7, 13-14, 8-9, 31-32, 24-28
Final using standard ACO [17]	139.5	6-7, 13-14, 8-9, 31-32, 24-28
Final using Branch exchange [15]	147.89	7-20, 8-14, 10-11, 30-31, 27-28
Final using Branch and Bound [16]	140.28	6-7,13-14, 9-10, 31-32, 24-28



**Fig. 3.** Initial configuration of the 118-bus system

**Table 3.** convergence of the standard ACO for the 32-bus system as in [17]

Iteration no	5	10	15	20	25	27
Power losses (KW)	145.8	144.3	144.3	142.6	140.1	139.5

The second test system is an 11KV system with one supply point, 118 bus and 15 tie lines whose data are given in [6]. The total substation loads for the initial configuration which is shown in Fig. (3) are 22709.7 KW and 17042.2 KVAR and the total power loss is 1294.68 KW. The HC-ACO parameters used are  $N=20$ ,  $\alpha=0.1$ ,  $\beta=0.9$ ,  $\rho=0.01$ ,  $\tau_0=1$ ,  $H_{max}=100$  and  $W_{max}=10$ . The final configuration which is shown in Fig. (4) was reached after 15 iterations with a power loss of 865.32 KW and 33.1% reduction in losses. Fig. (5) shows the improvement of the voltage profile after reconfiguration such that before reconfiguration the lowest bus voltage was 0.8685 PU while after reconfiguration the lowest bus voltage is 0.933 PU with 6.9% improvement.

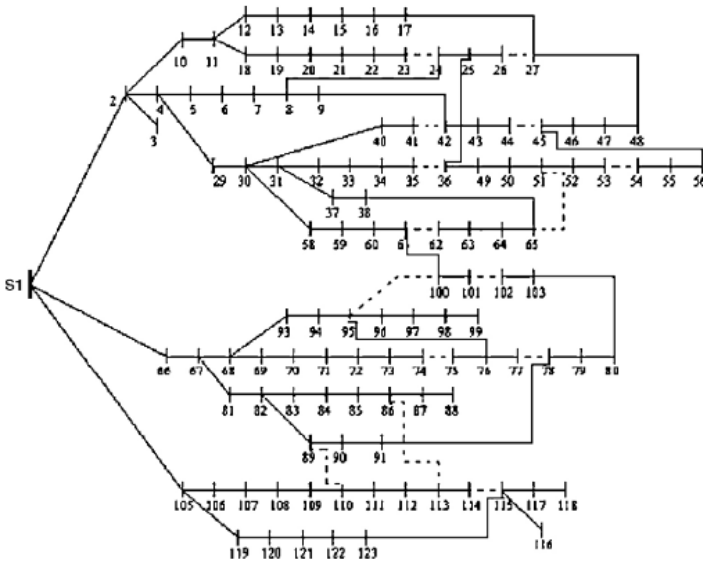


Fig. 4. Final configuration of the 118-bus system

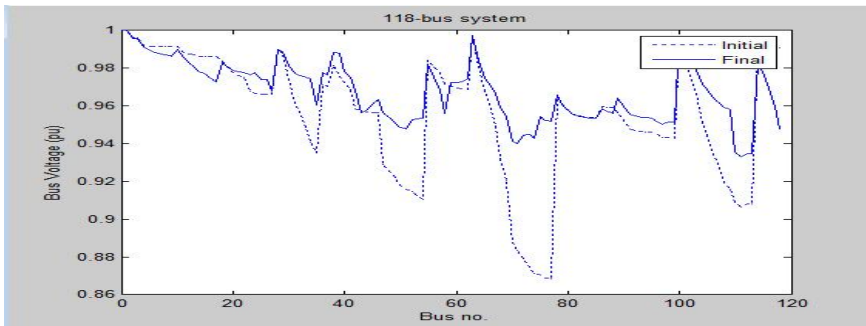


Fig. 5. Voltage profile of the 118-bus system before and after reconfiguration

Tables (2) and (4) show the results of reconfiguration of the 2 test systems using the HC-ACO and other algorithms from literature respectively. It is observed that the HC-ACO provides the same final configuration of AI based algorithms such as the standard ACO, Simulated Annealing (SA), Tabu Search (TS) and Particle Swarm (PS) with equal power losses. The HC-ACO also obtained better results than that obtained by purely heuristic algorithms such as the branch exchange method of Baran and Wu [15] and the branch and bound method of Shirmohammadi et al. [16] which prove the effectiveness of the proposed approach. Table (3) shows the evolution of the standard ACO algorithm for the 32-bus system as given in [17]. In comparison with Table (1), it is clear that implementing ACO in the HC framework comes with the benefit of scaling objective function value allowing rapid discovery of good solutions and fast optimum convergence.

**Table 4.** Results of reconfiguration of the 118-bus system

Reconfiguration	Losses(KW)	Tie Switches
Initial	1294.68	48-27,17-27,8-24, 56-45,65-51,38-65,9-42,61-100, 76-95,91-78,103-80,113-86, 110-89, 115-123, 25-36
Final using HC-ACO	865.322	45-44, 27-26, 22-23, 54-53, 51-50, 64-65, 41-42, 61-100, 76-77, 74-75, 80-79, 85-86, 89-110, 114-115, 33-34
Final using MTS [5] and ITS [6]	865.322	45-44, 27-26, 22-23, 54-53, 51-50, 64-65, 41-42, 61-100, 76-77, 74-75, 80-79, 85-86, 89-110, 114-115, 33-34
Final using Branch exchange [15]	885.56	45-44,17-27, 23-24, 53-52, 51-50, 64-65, 41-42, 61-100, 76-77, 74-75, 79-80, 85-86, 89-110, 114-115, 35-36

## 6 Conclusion

This paper presents the new Hyper-Cube framework for implementing the Ant Colony Optimization algorithm to solve the minimum loss distribution system reconfiguration problem. The new framework aimed at changing the pheromone updating rules to increase the robustness of the ACO algorithm making it easier the handling of the pheromone trails. The ACO applies three main rules that make the search more directed. It is a greedy search algorithm that makes use of positive feedback and constructive heuristic information to avoid premature convergence and enhance the capability of finding the optimal solution in the problem solving process resulting in an extremely powerful method for optimization problems. The validity and effectiveness of the HC-ACO were demonstrated by applying it to 32-bus and 118-bus systems. The HC-ACO proves to be useful for analyzing existing systems, helps in planning a future system, and is especially suitable for a large-scale practical system. Future studies can further demonstrate the effectiveness of the proposed algorithm by applying it larger practical distribution systems and distribution systems with Distributed Generations.

## References

1. Merlin, Back, H.: Search for a minimal-loss operating spanning tree configuration in an urban power distribution system. In: Proc. 5th Power System Computation Conf. (PSCC), Cambridge, UK, pp. 1–18 (1975)
2. Enacheanu, B., Raison, B., Caire, R., Devaux, O., Bienia, W., Hadjsaid, N.: Radial network reconfiguration using genetic algorithm based on the matroid theory. *IEEE Trans. Power Syst.* 23(1), 186–195 (2008)
3. Abdelaziz, A.Y., Mohamed, F.M., Mekhamer, S.F., Badr, M.A.L.: Distribution systems reconfiguration using a modified particle swarm optimization algorithm. *Elec. Power Sys. Res.* 79(11), 1521–1530 (2009)
4. Jeon, Y.-J., Kim, J.-C.: Application of simulated annealing and tabu search for loss minimization in distribution systems. *Int. J. Elec. Power & Energy Sys.* 26(1) (January 2004)
5. Abdelaziz, A.Y., Mohamed, F.M., Mekhamer, S.F., Badr, M.A.L.: Distribution system reconfiguration using a modified Tabu Search algorithm. *Elec. Power Syst. Res.* 80(8), 943–953 (2010)
6. Zhang, D., Fu, Z., Zhang, L.: An improved TS algorithm for loss-minimum reconfiguration in large-scale distribution systems. *Elec. Power Syst. Res.* 77, 685–694 (2007)
7. Sarfi, R.J., Salama, M.M.A., Chikhani, A.Y.: A survey of the state of the art in distribution system reconfiguration for system loss reduction. *Electric Power Systems Research* 31, 61–70 (1994)
8. Dorigo, M.: Optimization, Learning and Natural Algorithms (in Italian) Ph.D. dissertation, Dipartimento di Elettronica, Politecnico di Milano, Milan, Italy (1992)
9. Shirmohammadi, D., Hong, H.W.: A compensation-based power flow method for weakly meshed distribution and transmission networks. *IEEE Trans. Power Syst.* 3 (1988)
10. Carpaneto, E., Chicco, G.: Distribution system minimum loss reconfiguration in the Hyper-Cube Ant Colony Optimization framework. *Elec. Power Syst. Res.* 78, 2037–2045 (2008)
11. Blum, C., Dorigo, M.: The hyper cube framework for ant colony optimization. *IEEE Trans. Syst. Man Cyber.* 34(2) (April 2004)
12. Carpaneto, E., Chicco, G.: Ant-Colony Search-Based Minimum Losses Reconfiguration of Distribution Systems. In: IEEE MELECON, Dubrovnik, Croatia, pp. 971–974 (2004)
13. Daniel, C., Hafeezulla Khan, I., Ravichandran, S.: Distribution Network Reconfiguration For Loss Reduction Using Ant Colony System Algorithm. In: IEEE Indicon Conference, India, pp. 619–622 (2005)
14. Ahuja, A., Pahwa, A.: Using Ant Colony Optimization for Loss Minimization in Distribution Networks. In: Proceedings of the 37th Annual North American Power Symposium, pp. 470–474 (2005)
15. Baran, M.E., Wu, F.F.: Network reconfiguration in distribution systems for loss reduction and load balancing. *IEEE Trans. Power Deliv.* 4, 1401–1407 (1989)
16. Shirmohammadi, D., Hong, H.W.: Reconfiguration of electric distribution networks for resistive line loss reduction. *IEEE Trans. Power Deliv.* 4, 1492–1498 (1989)
17. Ghorbani, M.A., Hosseinian, S.H., Vahidi, B.: Application of Ant Colony System algorithm to distribution networks reconfiguration for loss reduction. In: 11th International Conference on Optimization of Electrical and Electronic Equipment, OPTIM, pp. 269–273 (May 2008)



# Bacterial Foraging Optimization Algorithm Trained ANN Based Differential Protection Scheme for Power Transformers

M. Geethanjali, V. Kannan, and A.V.R. Anjana

EEE Department, Thiagarajar College of Engineering, Madurai, India  
mgee@tce.edu, kannanvenkat@yahoo.com,  
avr\_tamil@yahoo.co.in

**Abstract.** To avoid the malfunction of the differential relay, alternate improved protection techniques are to be formulated with improved accuracy and high operating speed. In this paper an entirely new approach for detection and discrimination of different operating and fault conditions of power transformers is proposed. In the proposed scheme Artificial Neural Network (ANN) techniques have been applied to power transformer protection to distinguish internal faults from normal operation, magnetizing inrush currents and external faults. Conventionally Levenberg-Marquardt learning rule based back propagation (BP) algorithm is used for optimizing the weights and bias values of the neural network. In this paper bacterial foraging algorithm (BFA), based on the self adaptability of individuals in the group searching activities is used for adjusting the weights and bias values in BP algorithm instead of Levenberg-Marquardt learning rule.

## 1 Introduction

Protection of large power transformers is the most challenging problem in the area of power system relaying. [1-4]. In conventional differential protection technique based on the second harmonic restraint will have difficulty in distinguishing between an internal fault and an inrush current thereby threatening transformer stability [5,6]. To avoid the malfunction of the differential relay, alternate improved protection techniques for accurate and efficient discrimination between internal faults and inrush currents are to be formulated [5,7,8,9].

Recently, Artificial Neural Network (ANN) techniques have been applied to power transformer protection to distinguish internal faults from normal operation, magnetizing inrush currents and external faults [1,4]. In recent years bacterial foraging algorithm (BFA), based on the self adaptability of individuals in the group searching activities has attracted a great deal of interests in optimization field [9].

In this paper an entirely new approach for detection and discrimination of different operating and fault conditions of power transformers is proposed. In this approach BFA is used for adjusting the weights and bias values in BP algorithm instead of Levenberg-Marquardt learning rule. A suitable transformer model is required to

characterize the different operating conditions of power transformer. The simulation of power transformer for various operating conditions is performed using ATP (Alternate Transient Program) simulation tool in this paper. The current waveforms are obtained from ATP and fed to MATLAB as input. Diagnosis and analysis of these current waveforms are performed in MATLAB 7.0. It is concluded that the proposed scheme provides promising security, ability to not trip when it should not, dependability (ability to trip when it should) and speed of operation (short fault clearing time).

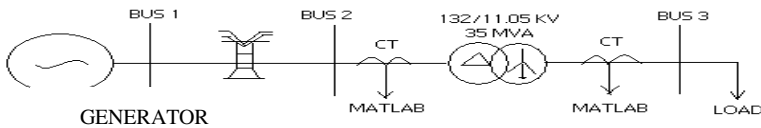
## 2 Modeling and Simulation

### 2.1 ATP Modeling of a Power Transformer

Real time model of a Power transformer for doing test and analysis is very expensive and time consuming. Development and validation of algorithms for a digital differential transformer protection require the preliminary determination of the power transformer model which simulates current signals for energization, over excitation, external fault, and internal fault conditions. Out of the various transformer models for simulation of low-frequency transients BCTRAN model in ATP is most suitable for this study and hence it is preferred.

### 2.2 Simulation of Different Power Transformer Operating Conditions Using ATP

For a two-winding three-phase power transformer, the modified BCTRAN model, which is a well known subroutine for a transformer model on ATP/EMTP with a nonlinear inductance included in the winding which is nearer to the core is used for simulation studies of power transformer. Figure 1 shows the system used in simulation studies.

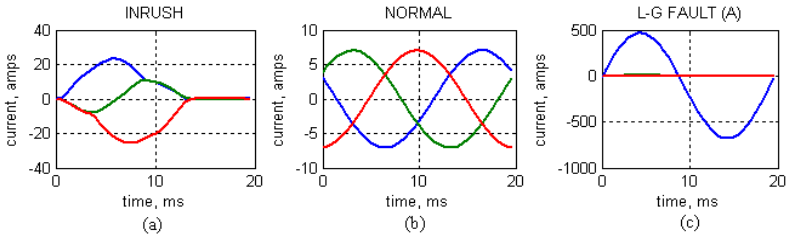


**Fig. 1.** Test system used for simulation

A three phase, two winding, 50 Hz, shell type power transformer rated as 35 MVA, 132/11.05 KV, with D/Y connection on HV and LV sides respectively, was employed in simulation and the test data are as follows:

<i>OC test</i>	: Voltage 100%,	Current 0.1316%,	Losses 18.244 kW.
<i>SC test</i>	: Impedance 26.291%,	Power 35 MVA,	Losses 192.53 kW.

It can be seen that the transformer as a step down transformer is connected between two sub transmission sections. The primary and secondary current waveforms, then, can be simulated using ATP for different operating and fault conditions, and these waveforms are brought into MATLAB for further investigations. For instances the differential currents of magnetizing inrush, normal operating and L-G fault at phase A are shown in figure 2.



**Fig. 2.** Simulation of differential current waveform for different operating conditions (a)→Magnetizing inrush(b)→Normal operation (c)→L-G fault at A

### 3 Bacterial Foraging Optimization

#### 3.1 Basic Concepts

Foraging can be modeled as an optimization process where an animal seeks to maximize energy obtained per unit time spent foraging. Search strategies form the basic foundation for foraging decisions. Animals search for and obtain nutrients in a way that maximizes

$$E/T \tag{1}$$

Where  $E$  is energy obtained and  $T$  is time spent for foraging (or they maximize long-term average rate of energy intake). Evolution optimizes foraging strategies since animals that have poor foraging performance do not survive.

#### 3.2 Bacterial Swarm Foraging for Optimization

In bacterial swarm foraging for optimization, the basic goal is to find the minimum of

$$J(\theta), \theta \in \mathfrak{R}^p \tag{2}$$

when we do not have the gradient  $\Delta J(\theta)$  Suppose  $\theta$  is the position of a bacterium and  $J(\theta)$  represents an attractant-repellant profile (i.e.. it represents where nutrients and noxious substances are located so  $J < 0$ ,  $J = 0$  and  $J > 0$  represent the presence of nutrients, a neutral medium, and the presence of noxious substances, respectively).

### Bacterial Swarm Foraging Optimization Algorithm

For initialization, the suitable values for  $p$ ,  $S$ ,  $N_c$ ,  $N_s$ ,  $N_{re}$ ,  $N_{ed}$ , and  $p_{ed}$  are chosen and the  $C(i)$ ,  $i=1,2,\dots,S$ . If swarming is used the parameters of the cell-to-cell attractant functions are to be picked up. In this paper the parameters given above are used. Also, initial values for the  $\theta^i$ ,  $i=1,2,\dots,S$ , must be chosen. Choosing these to be in areas where an optimum value is likely to exist is a good choice. The algorithm that models bacterial population chemotaxis, swarming, reproduction, elimination, and dispersal is discussed in this paper (initially,  $j = k = l = 0$ ). For the algorithm, it is to be noted that updates to the  $\theta^i$  automatically result in updates to  $P$ . Clearly, we could have added a more sophisticated termination test than simply specifying a maximum number of iterations.

Steps involved are:

**Step 1:** Elimination-dispersal loop:  $l = l+1$

**Step 2:** Reproduction loop:  $k = k+1$

**Step 3:** Chemotaxis loop:  $j = j+1$

- a) For  $i = 1, 2, \dots, S$ , take a chemotactic step for bacterium  $i$  as follows
- b) Compute  $J(I,j,k,l)$ . Let  $J(i,j,k,l) = J(i,j,k,l) + J_{cc}(\theta^i(j,k,l),P(j,k,l))$  (i.e., add on the cell-to-cell attractant effect to the nutrient concentration)
- c) Let  $J_{last}=J(I,j,k,l)$  to save this value since we may find a better cost via a run.
- d) Tumble: Generate a random vector  $\Delta(i) \in \mathbb{R}^p$  with each element  $\Delta_m(i)$ ,  $m=1,2,\dots,p$ , a random number on  $[-1,1]$
- e) Move: Let

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \tag{3}$$

- f) Compute  $J(i,j+1,k,l)$ , and then let  $J(i,j+1,k,l) = J(i,j+1,k,l) + J_{cc}(\theta^i(j,k,l),P(j,k,l))$
- g) Swim (note that we use an approximation since we decide swimming behavior of each cell as if the bacteria numbered  $\{1,2,\dots,i\}$  have moved and  $\{i+1,i+2,\dots,S\}$  have not; this is much simpler to simulate than simultaneous decisions about swimming and tumbling by all bacteria at the same time)

- i) Let  $m = 0$  (counter for swim length)
  - ii) While  $m < N_s$  (if have not climbed down too long)
    - Let  $m = m+1$ .
    - If  $J(i,j+1,k,l) < J_{last}$  (if doing better), let  $J_{last} = J(i,j+1,k,l)$
- and let

$$\theta^i(j+1,k,l) = \theta^i(j+1,k,l) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \tag{4}$$

And use this  $\theta^i(j+1,k,l)$  to compute new  $J(i,j+1,k,l)$  using step 'f'.

- Else, let  $m = N_s$ . This is the end of the while statement

h) Go to next bacterium ( $i + 1$ ) if  $i \neq S$  (i.e., go to b to process the next bacterium).

**Step 4:** If  $j < N_c$ , go to step 3. In this case, continue chemotaxis, since the life of the bacteria is not over

**Step 5:** Reproduction:

a) For the given  $k$  and  $l$ , and for each  $i = 1, 2, \dots, S$ , Let

$$J_{health}^i = \sum_{j=1}^{N_{c+1}} J(i, j, k, l). \quad (5)$$

be the health of bacterium  $i$  (a measure of how many nutrients it got over its lifetime and how successful it was at avoiding noxious substances). Sort bacteria and chemotactic parameters  $C(i)$  in order of ascending cost  $J_{health}$  (higher cost means lower health).

b) The  $S_r$  bacteria with the highest  $J_{health}$  values die and the other  $S_r$  bacteria with the best values split (and the copies that are made are placed at the same location as their parent).

**Step 6:** If  $k < N_{re}$ , go to step 2. In this case, if number of specified reproduction steps, are not reached the next generation in the chemotactic loop is started.

**Step 7:** Elimination-dispersal: For  $i = 1, 2, \dots, S$  with probability  $p_{ed}$ , eliminate and disperse each bacterium (this keeps the number of bacteria in the population constant). To do this, if you eliminate a bacterium, simply disperse one to a random location on the optimization domain.

**Step 8:** If  $l < N_{ed}$ , then go to step 1; otherwise end.

## 4 Design and Development of the Proposed Protective Relaying Scheme

### 4.1 Algorithm of the Proposed Protective Relaying Scheme

The proposed algorithm can be explained as follows:

- Sampled values of current signals are obtained from power transformer modeling with ATP for normal, inrush, internal and external fault conditions.
- Three phase differential currents ( $I_d$ ) are calculated for the operating and fault conditions.
- The differential current is checked for whether it exceeds the threshold value
- If the differential current exceeds threshold value then it is fed to ANN/BFANN. First of all the ANN architecture was trained using BP algorithm. Then the same architecture was trained using bacterial foraging optimization algorithm trained ANN (BFANN).
- The neural network architecture has four outputs which discriminates normal, inrush, internal and external fault conditions as follows:

- Normal operating condition - 0000
- Inrush condition - 1111
- Internal faults- Faults inside the transformer protection zone
- Line to Ground fault (Phase A) - 1000
- Line to Ground fault (Phase B) - 0100
- Line to Ground fault (Phase C) - 0010
- Line to Line fault (Phase A-B) - 1100
- Line to Line fault (Phase A-C) - 1010
- Line to Line fault (Phase B-C) - 0110
- Three Line to Ground fault - 1110
- External faults – Faults outside transformer protection zone
- External fault conditions (all) - 0001

- For normal operating, inrush and external fault conditions there will be no trip signal. Trip signal is given for internal fault conditions and the type of fault is identified.

#### 4.2 Implementation of Proposed Protection Scheme Using MLFFNN Architecture

The input vector  $p$ , which is one training pattern for ANN learning, is comprised of current time response for all three phases so that just these are sampled every 0.5 ms (3 x 40 samplings). Totally 120 (40 samples per phase) samples per cycle was taken for each of the fault, normal operating and inrush conditions. A set of 137 input current patterns are given for training process. Each input vector ‘ $p$ ’ is assigned a target output. A three layer feed forward ANN with error back propagation training algorithm is used for training. There are two hidden layers and one output layer. The two hidden layers have ‘tansig’ transfer function .The algorithm used for this transfer function is as follows:

$$\text{tansig}(n) = 2 / (1 + \exp(-2 * n)) - 1 \tag{6}$$

where

- $n$  --  $S \times Q$  matrix of net input (column) vectors and returns each element of  $N$  squashed between -1 and 1
- $S$  -- Number of neurons
- $Q$  -- no. of input vectors

The number of neurons in the hidden layers is chosen by trial and error method. The output layer has a ‘purelin’ transfer function and 4 numbers of neurons. The algorithm used for this transfer function is as follows:

$$\text{purelin}(n) = n \tag{7}$$

Levenberg-Marquardt backpropagation (‘trainlm’) is the training rule applied. The training algorithm has a learning rate of 0.1 and a momentum constant of 0.9. Maximum number of epochs is 600 and performance goal is  $10^{-5}$ .

### 4.3 Implementation of Proposed Protection Scheme Using BFANN Architecture

In this case bacterial foraging trained neural network is used for training. The input vector  $p$ , is the same value which is considered for MLFFNN architecture. There are two layers in this BFANN with one hidden layer and one output layer. The transfer function used in the first layer is 'slogsig'.

The algorithm used for this transfer function is as follows:

$$\text{slogsig}(n) = 2. / (2 + \exp(-n)) - 1 . \quad (8)$$

There are 4 neurons in the output layer. The transfer function used in the second layer is 'satlin'. The algorithm used for this transfer function is as follows:

$$\text{satlin}(n) = 0, \text{ if } n \leq 0; n, \text{ if } 0 \leq n \leq 1; 1, \text{ if } 1 \leq n. \quad (9)$$

The learning algorithm is bacterial foraging optimization algorithm. The algorithm is explained in the section 4.1. Values of the parameters assigned in bacterial foraging optimization algorithm trained neural network are given as follows:  $S = 12$ ,  $N_c = 4$ ,  $N_s = 50$ ,  $N_{re} = 4$ ,  $N_{ed} = 2$ ,  $p_{ed} = 0.25$ ,  $S_r = S/2$ ,  $C(i) = 0.1$ ,  $i = 1, 2, \dots, S$ . The bacteria are initially spread randomly over the optimization domain  $[-1, 1]$ .

## 5 Results and Discussion

The training was performed with a Pentium IV computing system having 512 MB RAM and 2.88 GHz processor speed.

### 5.1 MLFFNN Training Details

No. of training data	: 137
No. of test data	: 42
No. of training epochs (set)	: 600
No. of input samples	: (3*40) samples/cycle for 3 phases
First hidden layer	: 8 neurons (transfer function - tansig)
Second hidden layer	: 5 neurons (transfer function - tansig)
Output layer	: 4 neurons (transfer function - purelin)
Learning rate	: 0.1
Momentum constant	: 0.9
Performance goal	: 1e-5
No. of training epochs (converged)	: 600
Profile Time	: 1095 sec

### 5.2 BFANN Training Details

No. of training data	: 137
No. of test data	: 42
No. of training epochs (set)	: 600
No. of input samples	: (3*40) samples/cycle for 3 phases
Hidden layer	: 8 neurons (transfer function – slogsig)
Output layer	: 4 neurons (transfer function – satlin)
Performance goal	: 0
No. of training epochs (converged)	: 69
Profile Time	: 276 sec

### 5.3 Analysis of Results

With MLFFNN architecture it was found that the simulation time taken for fault detection and condition monitoring was 1095 seconds and it takes 600 epochs for convergence. The mean average values of training and testing error for MLFFNN architecture were 0.0013665 and 0.0029475 respectively. Some of the simulation results obtained during testing of ANN using BP algorithm are shown in table 1.

But when BFANN architecture was used for training, the training time taken for fault detection and condition monitoring is just 276 seconds and converges in 69 epochs itself. Moreover the accuracy is 100%. The same computing system was used for the training of this BFANN network too. Some of the simulation results obtained during training and testing of ANN using Bacterial foraging algorithm are shown in table 1.

**Table 1.** ANN Testing results using BP algorithm

Conditions	Output during testing of MLFFNN architecture								Average Error
	A1	T1	A2	T2	A3	T3	A4	T4	
<b>Normal</b>	0.0015	0	-0.0002	0	-0.0001	0	0.0008	0	0.00065
<b>Inrush</b>	0.9936	1	1.0148	1	0.9822	1	1.0103	1	0.012325
<b>LG Ph A</b>	1.0030	1	0.0070	0	0.0031	0	-0.0017	0	0.0037
<b>LG Ph B</b>	-0.0011	0	1.0000	1	-0.0002	0	-0.0006	0	0.000475
<b>LG Ph C</b>	-0.0007	0	-0.0007	0	0.9986	1	0.0005	0	0.000825
<b>LL – AB</b>	1.0000	1	1.0004	1	0.0003	0	0.0003	0	0.000775
<b>LL – BC</b>	0.0013	0	0.9990	1	1.0008	1	0.0026	0	0.001425
<b>LL – AC</b>	0.9986	1	0.0002	0	1.0003	1	-0.0007	0	0.00065
<b>3L-G</b>	0.9986	1	1.0014	0	1.0028	1	-0.0011	0	0.001675
<b>External Fault (all)</b>	-0.0146	0	-0.0003	0	0.0029	0	1.0101	1	0.006975



Figure 3 depicts convergence curve of MLFFNN in terms of number of epochs and error. Figure 4 depicts convergence curve of BFANN network in terms of number of epochs and error.

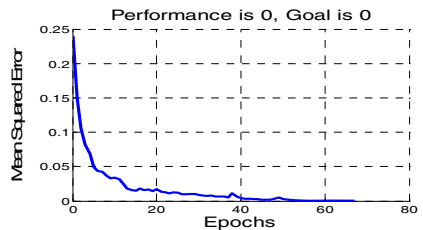
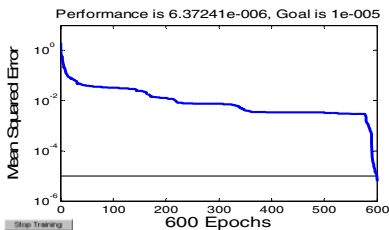
**Table 2.** ANN Training and Testing results using Bacterial foraging algorithm

Conditions	Output during training and testing of BFANN architecture								Average Error
	A1	T1	A2	T2	A3	T3	A4	T4	
Normal	0	0	0	0	0	0	0	0	0
Inrush	1	1	1	1	1	1	1	1	0
LG Ph A	1	1	0	0	0	0	0	0	0
LG Ph B	0	0	1	1	0	0	0	0	0
LG Ph C	0	0	0	0	1	1	0	0	0
LL – AB	1	1	1	1	0	0	0	0	0
LL – BC	0	0	1	1	1	1	0	0	0
LL – AC	1	1	0	0	1	1	0	0	0
3L-G	1	1	0	0	1	1	0	0	0
External Fault (all)	0	0	0	0	0	0	1	1	0

A1, A2, A3, A4 → Actual outputs T1, T2, T3, T4 → Target outputs

**Table 3.** Performance comparison of MLFFNN and BFANN

NETWORK	NO. OF EPOCHS	TRAINING TIME	MEAN SQUARED ERROR	ACCURACY
MLFFNN	600	1095 sec	0.0013665	99.99%
BFANN	69	276 sec	0	100%



**Fig. 3.** ANN training record obtained with Back Propagation algorithm

**Fig. 4.** ANN training record obtained with BFA

## 6 Conclusion and Future Work

In this paper two different ANN architectures have been used using different learning algorithms (BP algorithm and Bacterial Foraging algorithm). Both the architectures were trained for the different transformer operating and fault ( normal, inrush, internal and external fault) conditions discussed above.

First one is MLFFNN using BP algorithm and the second one is BFANN (ANN trained with Bacterial Foraging algorithm). In general, MLFFNN is used in power transformer protection field [1], [4], [9] which gives better results. But in this present work, from table (3), it is clear that the BFANN converges very quickly with high accuracy (100%) as compared to MLFFNN. Also it is seen that the network architecture is very simple which has two layers. The BFANN has been trained for all the possible sets of simulated data under different operating conditions of transformer. Thus the proposed BFANN based differential relaying for power transformer shows promising security, ability to not trip when it should not, dependability (ability to trip when it should) and speed of operation (short fault clearing time). In this proposed method, bacterial foraging optimization is used to adjust the weights and bias values to an optimum value and ANN is trained with these weights and bias values for identification and discrimination of faults. Thus this paper presents a novel technique to accurately monitor and discriminate different operating conditions of the power transformer (normal, inrush current, internal fault and external fault), detects the fault and issues a trip signal by using bacterial foraging optimization trained artificial neural network (BFANN).

In future this work shall be extended for different parts of the power systems such as complicated transmission lines with compensating devices including different fault parameters like fault inception angle and auto-reclosure condition.

**Acknowledgment.** The authors are thankful to the authorities of Thiagarajar College of Engineering, Madurai-625 015, for providing all the facilities to do this research work.

## References

1. Perez, L.G., Flechsig, A.J., Meador, J.L., Obradovic, Z.: Training an artificial neural network to discriminate between magnetizing inrush and internal faults. *IEEE Transactions on Power Delivery* 9(1), 434–441 (1994)
2. Bastard, P., Meunier, M., Regal, H.: Neural network-based algorithm for power transformer differential relays. *IEE Proceedings- Generation, Transmission and Distribution* 142(4), 386–392 (1995)
3. Zhang, Y., Ding, X., Liu, Y., Griffin, P.J.: An Artificial Neural Network Approach to Transformer Fault Diagnosis. *IEEE Transactions on Power Delivery* 11(4), 1836–1841 (1996)
4. Philer, J., Grcar, B., Dolinar, D.: Improved operation of power transformer protection using artificial neural network. *IEEE Transactions on Power Delivery* 12(3), 1128–1136 (1997)

5. Passino, K.M.: Distributed optimization and control using only a germ of intelligence. In: Proceedings of the 2000 IEEE International Symposium on Intelligent Control, July 17-19, pp. P5-P13 (2000)
6. Mao, P.L., Aggarwal, R.K.: A Novel Approach to the Classification of the Transient Phenomena in Power Transformers Using Combined Wavelet Transform and Neural Network. *IEEE Transactions on Power Delivery* 16(4), 654-660 (2001)
7. Passino, K.M.: Biomimicry of Bacterial Foraging for Distributed Optimization and Control. *IEEE Control Systems Magazine* 22(3), 52-67 (2002)
8. Geethanjali, M., Slochanal, S.M.R., Bhavani, R.: A novel approach for power transformer protection based upon combined wavelet transform and neural networks (WNN). In: The 7th International Power Engineering Conference, IPEC 2005, 29 November-2 December (2005)
9. Segatto, E.C., Cury, D.V.: A Differential Relay for Power Transformers Using Intelligent Tools. *IEEE Transactions on Power Systems* 21(3) (August 2006)
10. Geethanjali, M., Mary Raja Slochanal, S., Bhavani, R.: PSO-Trained ANN Based Differential Protection Scheme For Power Transformers. *Neuro Computing* 71, 904-918 (2008)
11. Geethanjali, M., Mary Raja Slochanal, S., Bhavani, R.: A combined WNN approach for discrimination between Magnetizing Inrush current and Internal fault in a power transformer. *IEEMA Journal*, 62-65 (February 2006)
12. <http://www.eeug.org>
13. <http://www.mathworks.com>

# Reduced Order Modeling of Linear MIMO Systems Using Soft Computing Techniques

Umme Salma<sup>1</sup> and K. Vaisakh<sup>2</sup>

<sup>1</sup>Department of Electrical and Electronics Engineering, GITAM Institute of Technology,  
GITAM University, Visakhapatnam, AP, India  
usalma123@gmail.com

<sup>2</sup>Department of Electrical Engineering, AU College of Engineering, Andhra University,  
Visakhapatnam, AP, India  
vaisakh\_k@yahoo.co.in

**Abstract.** A method is proposed for model order reduction for a linear multivariable system by using the combined advantages of dominant pole reduction method and Particle Swarm Optimization (PSO). The PSO reduction algorithm is based on minimization of Integral Square Error (ISE) pertaining to a unit step input. Unlike the conventional method, ISE is circumvented by equality constraints after expressing it in frequency domain using Parseval's theorem. In addition to this, many existing methods for MIMO model order reduction are also considered. The proposed method is applied to the transfer function matrix of a 10<sup>th</sup> order two-input two-output linear time invariant model of a power system. The performance of the algorithm is tested by comparing it with the other soft computing technique called Genetic Algorithm and also with the other existing techniques.

**Keywords:** Reduced order model, Integral Square Error, Parseval's theorem, Particle Swarm Optimization, Genetic Algorithm.

## 1 Introduction

In real problems the analysis of high order systems (HOS) is costly and tedious. Hence simplification procedure for original HOS are generally employed to realize for simple models based on physical considerations or by using mathematical approaches. Numerous methods are available in the literature for order reduction for linear continuous systems in time domain as well in frequency domain such as step response, frequency response etc. The reduced order model must be a good approximation of original model and it should retain the physical characteristics of the system such as step response, stability etc. Further, numerous methods of order reduction are also available in the literature, which are based on the minimization of the integral square error (ISE) criterion. However, a common feature in these methods is that the values of the denominator coefficients of the low order system (LOS) are chosen arbitrarily by some stability preserving methods such as dominant pole, Routh Approximation Methods, Routh Stability Criterion etc. and then the numerator coefficients of the LOS are determined by minimization of the ISE.

In the recent years, Particle Swarm Optimization (PSO) technique appeared as a promising algorithm for handling the optimization problems. Particle swarm optimization (PSO) is a population based stochastic optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995, inspired by social behavior of bird flocking or fish schooling [1]. PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithm (GA) with a population of random solutions and searches for optima by updating generations. Unlike GA, however, PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles. Here PSO method for model reduction is proposed. PSO method is based on minimization of Integral square error (ISE) between the transient responses of original higher order model and reduced order model pertaining to a unit step input [2].

Another approach for calculation of ISE [3] is considered. Unlike the conventional method, ISE is alternatively expressed in frequency domain using Parseval’s theorem and evaluated by considering a set of equality constraints involving the coefficients in the numerator and denominator of the original and reduced order transfer functions. Basically the method starts with the fixation of denominator coefficients of lower order system (LOS) by dominant pole method and determining the coefficients of numerator polynomials of each element of LOS transfer function matrix by minimizing the ISE in between the transient responses of original and LOS using PSO. The algorithm is described in detail in the following sections and is applied to a 10<sup>th</sup> order two input-two output linear time invariant practical power system. In the present paper, in addition to PSO and GA ten more existing methods for MIMO model reductions are also considered.

## 2 Problem Formulation

Let the transfer function of the higher order system (HOS) of order ‘r’ having ‘p’ inputs and ‘m’ outputs be

$$[G(s)] = \frac{1}{D(s)} \begin{bmatrix} a_{11}(s) & a_{12}(s) & a_{13}(s) & \dots & a_{1p}(s) \\ a_{21}(s) & a_{22}(s) & a_{23}(s) & \dots & a_{2p}(s) \\ \vdots & \vdots & \vdots & \dots & \vdots \\ a_{m1}(s) & a_{m2}(s) & a_{m3}(s) & \dots & a_{mp}(s) \end{bmatrix} \tag{1}$$

The general form of  $g_{ij}(s)$  of  $[G(s)]$  is taken as:

$$g_{ij}(s) = \frac{a_{ij}(s)}{D(s)} = \frac{a_0 + a_1s + a_2s^2 + \dots + a_{n-1}s^{n-1}}{b_0 + b_1s + b_2s^2 + \dots + b_{n-1}s^{n-1} + s^n} \tag{2}$$

$$(or) \quad g_{ij}(s) = \frac{a_0 + a_1s + a_2s^2 + \dots + a_{n-1}s^{n-1}}{(s + \lambda_1)(s + \lambda_2) \dots (s + \lambda_n)} \tag{3}$$

Where  $-\lambda_1 < -\lambda_2 < \dots < -\lambda_n$  are poles of the HOS.

Let the transfer function matrix of the LOS of order ‘r’ having ‘p’ inputs and ‘m’ outputs to be synthesized is:

$$[G(s)] = \frac{1}{\bar{D}(s)} \begin{bmatrix} b_{11}(s) & b_{12}(s) & b_{13}(s) & \dots & b_{1p}(s) \\ b_{21}(s) & b_{22}(s) & b_{23}(s) & \dots & b_{2p}(s) \\ \vdots & \vdots & \vdots & \dots & \vdots \\ b_{m1}(s) & b_{m2}(s) & b_{m3}(s) & \dots & b_{mp}(s) \end{bmatrix}$$

Or  $[R(s)] = [r_{ij}(s)]$ ,  $i = 1,2, \dots, m; j = 1,2, \dots, p$  (4)

The general form of  $[r_{ij}(s)]$  of  $[R(s)]$  is taken as

$$r_{ij}(s) = \frac{b_{ij}(s)}{\bar{D}(s)} = \frac{\alpha_0 + \alpha_1s + \alpha_2s^2 + \dots + \alpha_{n-1}s^{r-1}}{d_0 + d_1s + d_2s^2 + \dots + d_{n-1}s^{r-1} + s^r}$$
 (5)

or  $g_{ij}(s) = \frac{\alpha_0 + \alpha_1s + \alpha_2s^2 + \dots + \alpha_{n-1}s^{r-1}}{(s+\lambda_1)(s+\lambda_2)\dots(s+\lambda_r)}$  (6)

Where  $-\lambda_1 < -\lambda_2 < \dots < -\lambda_r$  are the dominant poles of the HOS.

Depending on the order to be reduced to, the poles nearest to the origin are retained. Therefore the denominator polynomial is given by [4]:

$$\bar{D}(s) = d_0 + d_1s + d_2s^2 + \dots + d_{n-1}s^{r-1} + s^r$$
 (7)

### 3 Particle Swarm Optimization

The PSO method is a population based search algorithm where each individual is referred to as particle and represents a candidate solution. Each particle treated as a point in a d-dimensional space and represented as  $X = (x_{i1}, x_{i2}, \dots, x_{id})$  and flies through the search space with an adaptable velocity that is dynamically modified according to its own flying experience and also the flying experience of the other particles. Further, each particle has a memory and hence capable of remembering the best position in the search space ever visited by it. The best previous position of particle that corresponds with the fitness value represented as  $pbest = (p_{i1}, p_{i2}, \dots, p_{id})$  and the best position of all particles in the population is denoted as  $gbest$ . In each iteration the value of  $gbest$  and  $pbest$  are calculated. For the  $n^{th}$  iteration velocity and particle's position are updated as shown in the following equations respectively.

$$v_{id}^{n+1} = wv_{id}^n + c_1r_1^n(P_{id}^n - x_{id}^n) + c_2r_2^n(P_{id}^n - x_{id}^n)$$
 (8)

$$x_{id}^{n+1} = x_{id}^n + v_{id}^{n+1}$$
 (9)

Where,  $p_{id}$  =  $pbest$  of particle I,  $p_{gd}$  =  $gbest$  of the group,  $w$  = inertia weight.  $c_1, c_2$  = cognitive and social acceleration respectively.  $r_1, r_2$  = random numbers uniformly distributed in the range (0, 1).

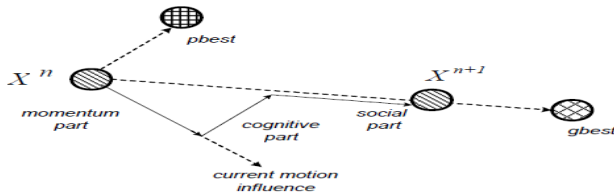


Fig. 1. Position updates in PSO for a two dimensional Parameters

In PSO, each particle moves in the search space with a velocity according to its own previous best solution and its group’s previous best solution. The velocity update in PSO consists of three parts; namely momentum, cognitive and social parts. The balance among these parts determines the performance of a PSO algorithm [2].

### 4 PSO Model Order Reduction

The various steps involved in PSO model reduction are as follows:

- Step1: Specify the parameters of PSO.
- Step2: Generate the initial population for the Particles.
- Step3: Find the fitness value ISE for the initial Population. Here objective function  $f(x)$  is fitness value ISE.
- Step4: The velocity and position of all particles are randomly set within pre-defined ranges.
- Step5: At each iteration, the velocities and positions of all particles are updated.
- Step6: Update the memory by updating  $p_{id}$  and  $p_{gd}$ . When condition is met  $p_{id} = p_i$  if  $f(p_i) > f(p_{id})$ ,  $p_{gd} = g_i$  if  $f(g_i) > f(p_{gd})$
- Step7: Stopping Condition – The algorithm repeats steps 5 to 7 until pre-defined number of iterations. It reports the values of  $g_i$ , and  $f(g_i)$  as its solution. This technique can be extended to multi-input and multi-output systems.

Parameters used for PSO algorithm are Swarm Size=10; Max. Generations=100;  $c_1 = 0.2$ ;  $c_2 = 0.2$ ;  $w_{start} = 0.9$ ;  $w_{end} = 0.4$ .

### 5 Integral Square Error Minimization Technique (ISE)

The ISE of the unit-step response is given by

$$ISE = \int_0^\infty [y(t) - y_r(t)]^2 dt \tag{10}$$

Where  $y(t)$  and  $y_r(t)$  denote the unit-step responses of original and reduced order systems respectively.

Using Parseval’s theorem the ISE can alternatively be expressed in the frequency domain and can be evaluated as described by [3]:

$$ISE = \frac{1}{(-1)^{p-1} 2|\Omega|} [h_0 c_{p-1} Q_{p-1} - h_0 \{c_{p-2} Q_{p-2} - c_{p-3} Q_{p-3} \dots + (-1)^{p-1} c_1 Q_1\} + (-1)^{p-1} c_0 Q_0] \tag{11}$$





$$\lambda_1 = -0.1001, \lambda_{2,3} = -0.2392 \pm j3.2348, \lambda_{4,5} = -0.8977 \pm j1.3552, \lambda_6 = -2.1375, \lambda_7 = -9.6454, \lambda_8 = -11.9632, \lambda_{9,10} = -19.0451 \pm j2.4859$$

The reduced transfer function matrix after applying the proposed algorithm is:

$$[R(s)] = \frac{1}{\tilde{D}(s)} \begin{bmatrix} b_{11}(s) & b_{12}(s) \\ b_{21}(s) & b_{22}(s) \end{bmatrix} \tag{13}$$

$$b_{11}(s) = 7.07s^2 - 18.24s - 2.49 \quad b_{21}(s) = -0.78s^2 + 7.9s + 1.2$$

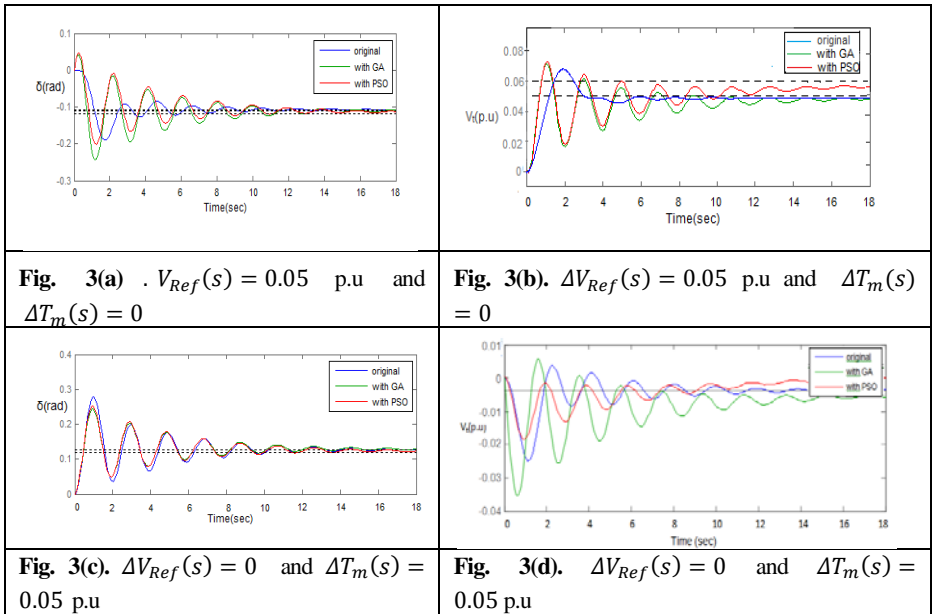
$$b_{12}(s) = 0.68s^2 + 29.97s + 2.5 \quad b_{22}(s) = -0.02s^2 - 2.4s + 0.03$$

Where  $\tilde{D}(s) = s^3 + 0.5785s^2 + 10.5690s + 1.0532$

The poles of the LOS  $[R(s)]$  are at  $\lambda_1 = -0.1001, \lambda_{2,3} = -0.2392 \pm j3.2348$

The adequacy of the 3<sup>rd</sup> order reduced models obtained above is tested by comparing with the original 10<sup>th</sup> order system and also for GA by Parmar [4] by time responses of the outputs (i.e.  $\delta$  and  $V_i$ ) for two distinct input Step changes in fig 3(a)-(b):

- With  $\Delta V_{Ref}(s) = 0.05$  p.u and  $\Delta T_m(s) = 0$
- With  $\Delta V_{Ref}(s) = 0$  and  $\Delta T_m(s) = 0.05$  p.u.



From the above simulation results, the time responses it is clear that the 3<sup>rd</sup> order reduced system obtained by the proposed algorithm is adequate coincide quite well with those of the original order system for the same input step change.

## 7 Comparisons of Reduced Order Models

Comparison of the proposed algorithm with some well known existing order reduction techniques are also shown in Table 1.

**Table 1.** Comparison of reduced order models

$r_{11}$	$r_{12}$	$r_{21}$	$r_{22}$
<b>I. Proposed Method PSO</b>			
$7.07s^2-18.24s-2.49$ ISE: 3.0328	$0.68s^2+29.97s+2.5$ ISE: 0.4272	$-0.78s^2+7.9s+1.2$ ISE: 0.6558	$-0.02s^2-2.4s+ 0.03$ ISE: 0.0580
Reduced Denominator: $s^3 + 0.5785s^2 + 10.5690s + 1.0532$			
<b>II. Parmar, G., et.al [4]</b>			
$7.4s^2 - 24s - 2.3$ ISE: 3.6563	$0.63s^2+28.9s+ 2.67$ ISE: 0.5287	$-0.6s^2+7.96s+1.03$ ISE: 0.7850	$-1.51s^2-2.99s-0.081$ ISE: 0.2263
Reduced Denominator: $s^3 + 1.877s^2 + 3.313s + 0.327$			
<b>III. Rama Jaya Lakshmi et.al. [5]</b>			
$-2.45s^2-2.24s-0.19$ ISE: 2.2542	$4.32s^2+2.79s+0.24$ ISE: 10.3816	$1.11s^2+1.01s+0.09$ ISE: 0.1102	$-0.38s^2-0.11s- 0.01$ ISE: 0.0980
Reduced Denominator: $s^3+1.453s^2+1.066s+0.092$			
<b>IV. Bei-bei, Wu., &amp; Chaun-qing, G.U. [6]</b>			
$-1.84s^2-6.94s-0.7$ ISE: 5.7421	$6.66s^2+8.97s+ 0.84$ ISE: 34.2366	$0.85s^2+3.13s+ 0.32$ ISE: 0.0946	$1.01s^2 - 0.36s-0.02$ ISE: 0.3224
Reduced Denominator: $s^3 + 1.877s^2 + 3.313s + 0.327$			
<b>V. Vishwakarma, C.B. and Prasad, R. [7]</b>			
$8.8s^2-64.35s-20.1$ ISE: 1.0575	$38.27s^2+88.78s+24$ ISE: -3.1511	$-3.03s^2+29.0s+ 9.1$ ISE: 0.1228	$7.27s^2-5.23s -0.72$ ISE: -0.0091
Reduced Denominator: $s^3 + 10.03s^2 + 32.28s + 9.374$			
<b>VI. Prasad, R. et.al. [8]</b>			
$-2.43s^2-2.24s-0.2$ ISE: $1.9304 \times 10^3$	$4.33s^2+2.79s+0.24$ ISE: $6.5014 \times 10^3$	$1.11s^2-1.01s-0.089$ ISE: 398.2047	$-0.39s^2-0.11s-0.03$ ISE: 58.4842
Reduced Denominator: $s^3+1.453s^2 + 1.066s + 0.093$			
<b>VII. Jayanta Pal and L. M. Ray [9]</b>			
$-1.83s^2-6.94s -0.7$ ISE: 68.3341	$6.66s^2+8.7s+ 0.838$ ISE: 34.2065	$0.85s^2+3.13s+ 0.32$ ISE: 0.0947	$1.01s^2-0.35s -0.03$ ISE: 0.3221
Reduced Denominator: $s^3 + 1.877s^2 + 3.312s + 0.327$			
<b>VIII. Shieh, L.S., and Wei, Y. J. [10]</b>			
$-2.22s^2-5.92s- 0.57$ ISE: 2.7234	$-6.45s^2+7.4s+0.68$ ISE: 13.5551	$1.02s^2+2.67s-0.256$ ISE: 0.0801	$-0.89s^2-0.3s- 0.02$ ISE: 0.1117
Reduced Denominator: $s^3 + 1.895s^2 + 2.822s + 0.264$			
<b>IX. Shamash, Y. [11]</b>			
$-2.22s^2-5.92s-0.57$ ISE: 2.7210	$6.45s^2+7.4s+ 0.677$ ISE: 13.5562	$1.02s^2-2.67s+0.256$ ISE: 0.0800	$-0.89s^2-0.2s-0.02$ ISE: 0.1117
Reduced Denominator: $s^3 + 1.895s^2 + 2.822s + 0.264$			

**Table 1. (Continued)**

<b>X. Nahid Habib et.al. [12]</b>			
- 64.36s-20.12 ISE: 1.7905	88.79s + 24.01 ISE: 1.5121	29.02s+ 9.07 ISE: 0.1844	- 5.2286s -0.7193 ISE: 0.0585
Reduced Denominator: $s^3 + 10.03s^2 + 32.28s + 9.374$			
<b>XI. Anurag Vijay Agrawal and Ankit Mittal [13]</b>			
13.2s <sup>2</sup> -175.9s-11.7 ISE: 1.9882	94.5s <sup>2</sup> + 217s +13.9 ISE: -1.5094	-5.54s <sup>2</sup> +79.4s+5.3 ISE: 0.2056	-22.4s <sup>2</sup> - 8.0s-0.42 ISE: 0.0038
Reduced Denominator: $s^3 + 19.26s^2 + 83.3s + 5.454$			
<b>XII. Liaw, C.M. [14]</b>			
5.76s <sup>2</sup> -22.13s+3.58 ISE: -2.2591	7.39s <sup>2</sup> +27.76s+ 2.69 ISE: 1.5106	-2.5s <sup>2</sup> +9.9s+1.02 ISE: 0.5484	-2.72s <sup>2</sup> -1.12s-0.08 ISE: 0.2481
Reduced Denominator: $s^3 + 0.5785s^2 + 10.57s + 1.053$			

An error index ISE known as Integral Square Error in between the transient parts of original and reduced  $r_{ij}(s)$  order models are calculated to measure the accuracy of the LOS. The smaller the ISE, the closer is  $r_{ij}(s)$  to  $g_{ij}(s)$ . In the above tabular form it has been observed that in most of the cases the value of ISE obtained from PSO is less with that of other methods. When compared with some methods regarding ISE values, the performance of PSO can be increased by the improvement of PSO variants, which is a future study of work.

## 8 Conclusion

In the present work, the author proposes an algorithm for multi-input multi-output model order reduction by PSO based on the minimization of Integral Square Error (ISE) pertaining to a unit step input. Here ISE is circumvented by equality constraints after expressing it in frequency domain using Parseval’s theorem. In the proposed method the denominator coefficients of the low order system (LOS) are preserved by dominant pole method and then the numerator coefficients of the LOS are determined by minimization of the ISE. The proposed method is applied to a 10<sup>th</sup> order two-input and two- output linear time invariant model of a practical power system. The adequacy of lower order models obtained by the proposed method has been judged by comparing the output time responses to the corresponding ones of the original system model. The reduced order models are compared with the Genetic Algorithm method. These methods are also compared for their ISE values with the other ten existing methods to prove the validity of the proposed method. Here PSO has been proved as a promising algorithm for handling MIMO model order reduction problems. In future study the concentration has to be done towards improvement of PSO variants to increase the efficiency of PSO.

## References

- [1] Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: IEEE Int. Conf. on Neural Networks, IV, pp. 1942–1948 (1995)
- [2] Pamar, G., Mukherjee, S., Prasad, R.: Relative mapping errors of linear time invariant systems caused by Particle swarm optimized reduced order model. *World Academy of Science and Technology*, 336–342 (2007)
- [3] Singh, V., Chandra, D., Kar, H.: Optimal Routh approximants through integral squared error minimization: Computer-aided approach. *IEE Proc., Contr. Theory Appl.* 151, 53–58 (2004)
- [4] Parmar, G., Mukherjee, S., Prasad, R.: Reduced order modeling of Linear MIMO systems using Genetic algorithm. *International Journal of Simulation Modeling* 6(3), 173–184 (2007)
- [5] Lakshmi, R.J., Mallikarjuna Rao, P., Vishnu Chakravarti, C.: A method for the reduction of MIMO systems using Interlacing property and Coefficients matching. *International Journal of Computer Applications* (0975 – 8887) 1(9), 14–17 (2010)
- [6] Bei-bei, W., Chaun-qing, G.U.: A matrix Pade type – Routh model reduction method for multivariable linear systems. *Journal of Shanghai University*, 377–380 (2006)
- [7] Vishwakarma, C.B., Prasad, R.: Order reduction using the advantages and differentiation method and factor division algorithm. *Indian Journal of Engineering and Material Sciences* 15, 447–451 (2008)
- [8] Prasad, R., Sharma, S.P., Mittal, A.K.: Improved Pade approximants for multivariable systems using Stability equation method. *Institution of Engineers India IE (I) Journal-EL* 84, 161–165 (2003)
- [9] Pal, J., Ray, L.M.: Stable Pade approximants to multivariable systems Using a mixed method. *Proceedings of the IEEE* 68(1) (1980)
- [10] Shieh, L.S., Wei, Y.J.: A mixed method for multivariable system reduction. *IEEE Trans. Automatic Control* AC-20, 429–432 (1975)
- [11] Shamash, Y.: Multivariable system reduction via modal methods and Pade approximation. *IEEE Transactions on Automatic Control* 20, 815–817 (1975)
- [12] Habib, N., Prasad, R.: An observation on the differentiation and modified Caueer continued fraction expansion approaches of model reduction technique. In: XXXII National Systems Conference (December 2008)
- [13] Agrawal, A.V., Mittal, A.: Reduction of large scale linear MIMO systems using Eigen Spectrum analysis and CFE form. In: XXXII National Systems Conference (December 2008)
- [14] Liaw, C.M.: Mixed method of model reduction for linear multivariable systems. *International Journal of Systems Science* 20(11), 2029–2041 (1989)

# Statistical and Fusion Based Hybrid Approach for Fault Signal Classification in Electromechanical System

Tribeni Prasad Banerjee and Swagatam Das

Electronics and Telecommunication Engineering Department,  
Jadavpur University, Kolkata-700032  
t\_p\_banerjee@yahoo.com,  
swagatamdas19@yahoo.co.in

**Abstract.** Motor fault diagnostics in dynamic condition is a typical multi-sensor fusion problem. It involves the use of multi-sensor information such as vibration, sound, current, voltage and temperature, to detect and identify motor faults. According to our experiments in BLDC motor controller results, the system has potential to serve as an intelligent fault diagnosis system in other hard real time system application. To make the system more robust we make the controller more adaptive that give the system response more reliable by the multisensory fusion techniques. We introduce a hybrid model based new methods and evaluate the performance of the proposed information fusion system. Finally, we report the efficiency of this system in dealing with controller stability and its nonlinear information that may arise among the sensors.

**Keywords:** Motor diagnosis, Information fusion, Sensor fusion, Support Vector Machine, Sort Term Fourier Transform, Brush less Direct Current Motor, signal classification, Fault Classifier.

## 1 Introduction

A notable research has tried form last two decade to overcome the problem of using multiple sensors to achieve better performance in diagnostic of high speed motor condition as well as predictive control of that system. The process of combining the provided information from multiple sensors is called sensor data fusion, and this technique can overcome a number of problems ranging from noise to incipient sensor failure. Even, one can increase the system's accuracy and the reliability using sensor fusion [1] absence of this issue also. The most conventional approach to sensor fusion includes: Kalman filtering, the weighted average, Bayesian estimators, adaptive observers, algebraic functions and nonlinear system fusion [2], [4], [5]. Our objective in this study is to propose a new hybrid approach for multiple sensor data fusion and motor fault detection. For this new approach, no previous knowledge is required about the sensors signals statistics, or the system behavior, and no learning or training processes are required. The work in this paper is continuing a previous work [6], proposing a new hybrid fusion system consists of the following four main phases:

- a) In the first phase the Signal will separated by STFT (Short Term Fourier Transform) [7], [8] by its frequency level and amplitude.
- b) The next phase the system faults are modelled as changes in the sensor gain with magnitude given by a nonlinear function of the measurable output and input signals. An adaptive time based observer is proposed in order to monitor the system for unanticipated sensor failures.
- c) Then a fusion block has been proposed which fused the sensor data.
- d) In this stage the information provided by the previous two steps and fused data helps to take decision SVM (Support Vector Machine) [9] based fault predictive system modelling.

In this paper, we proposed an incipient signal classification which is optimally separated by SVM and after classification of the signal or a optimal threshold value a controller strategy is implanted into a evolvable hardware which gives a hard and real time responsive or responsive system.

## 2 Principal of Sensor Fusion Architecture

The general fusion techniques have been shown in the Fig. 1. As shown the system consists of  $n$  input signals (from  $n$  sensors) and the objective is to achieve one fused output  $S_f$  based on these inputs. If the input sensors try to measure the same type of signal, i.e. light, sounds, temperature, position or velocity, the value measured by all the sensors at any time instant  $t$ , should be the same ideally, but due to the fault influence during the sensors operation such as parameter changes or changes in the sensors operational characteristics [10],[11], the values will be distributed in some manner relative to the acceptable measurements region, and the system is in a chaotic situation and to control this problem we proposed this hybrid techniques that measured values of sensor to the acceptable region of the signal range, while the other sensor possibility is the measured values of the acceptable region of fault or not; most of the time it is possible for getting the accuracy and better result for fused hybrid techniques than the single data fusion techniques as shown in the Fig. 4.

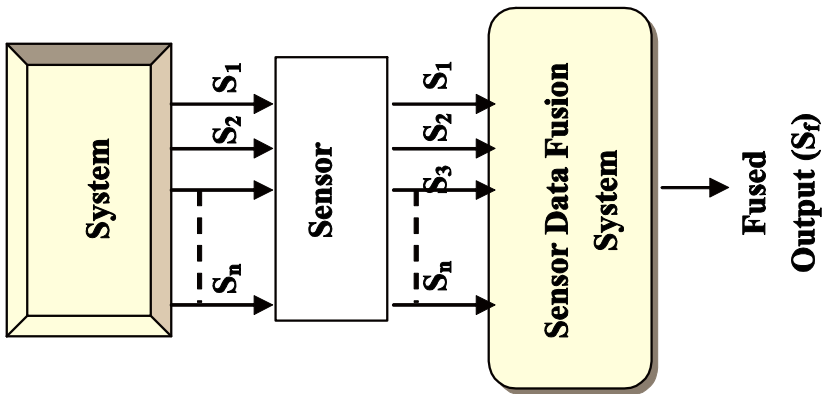


Fig. 1. Sensor Data acquisition and Fusion Techniques

### 3 Architecture of Real Time Dynamic Sensor Observer

Hard real time system is that system which has a critical time limit [12] or a reactive system where the system inputs are dynamically changes with time. Within that time duration the system has to response otherwise the system is going to a catastrophic loss or high probability for human death. The real time systems are basically two types one is reactive system and another is embedded system. A reactive system is always react with the environment (online aircraft valve signal monitoring from a actuator signal) and another is embedded system which is used to control specialized hardware that is installed within a larger system (such as a microprocessor that controls anti-lock brakes in an automobile) In our system is more reactive with the environments. Here in our proposed system the online dynamic continuous signals is comes and we capture the signal and transform it and pass through into our classifier and take the decision. The EFSM model [13] does the real time operation within a few micro seconds. This makes the system first responsive according to the decision of the output and the reactive system makes a more intelligent neuro adaptive system.

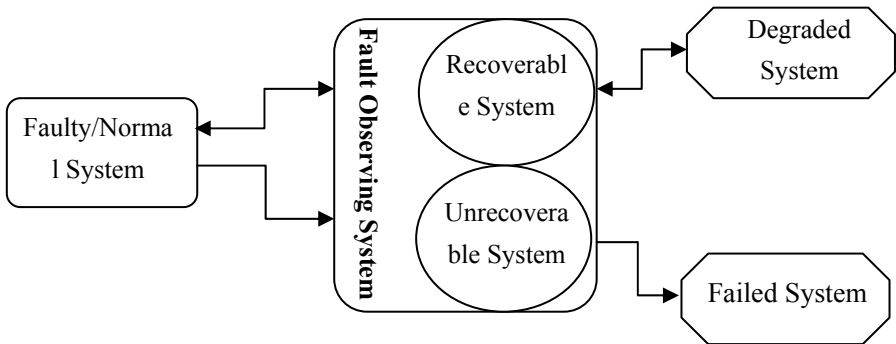


Fig. 2. State Model based Classification of Signal Condition Monitoring

### 4 Signal Preprocessing for STFT

Signal classification is a major research area in the real time DSP, because of the real time operation the classification of signal is very crucial issue. Even the signal classification has lots of problem, so we prefer the STFT. Here we briefly review the chaos detector based on the STFT proposed in [14]. The STFT is defined as The short-time Fourier transform (STFT), or alternatively short-term Fourier transform, is a Fourier-related transform used to determine the sinusoidal frequency and phase content of local sections of a signal as it changes over time.

$$STFT(t, f) = \int_{-\alpha}^{\alpha} x(t + \tau)w(\tau)e^{-j2\pi f\tau} d\tau \tag{1}$$

where  $x(t)$  is signal of interest (in this paper, it is a voltage or a current from the BLDC motor), and  $w(\tau)$  is the window function, where  $w(\tau)=0$  for  $|\tau|>T/2$  and  $T$  is window width. In order to avoid complex-valued STFT, we use its squared magnitude, i.e., the spectrogram  $SPEC(t, f) = |STFT(t, f)|^2$ .

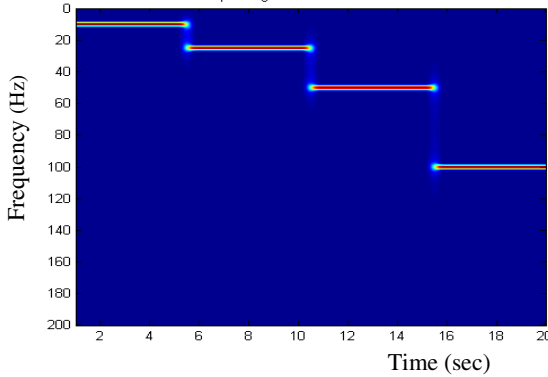
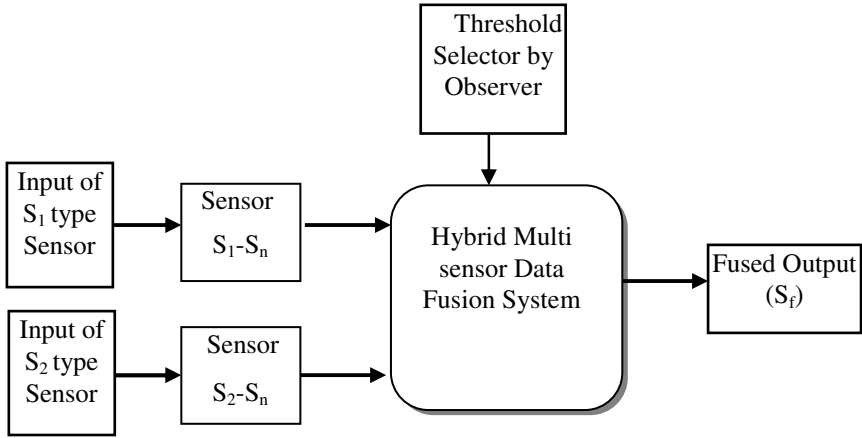


Fig. 3. Original Signal after the STFT transformed

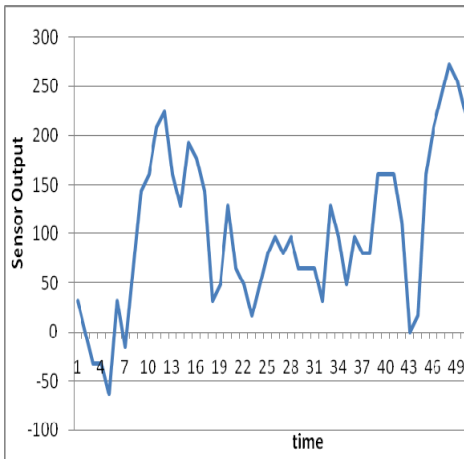
### 5 Architecture of the Hybrid Multi Sensor Data Fusion

The system called hybrid because the system working in various condition. The Probabilistic approaches and Machine learning both mechanisms gives their best output at time to time basis. Whenever required the signal classification takes into action. In this section we demonstrate how the proposed method can be used to resolve the decision conflict and helps to improve the performance of the fault signal classification than the previous SVM and STFT based methods [6]. From the architectural point of view we use two sensors to perform online inspection of motor current signal, rotation speed of the motor: an novelty detection sensor and tachometer. We collected a large number of motor current signals in a various condition. We then applied time domain, frequency domain and statistical signal processing methods to detect and classify faults. In a number of cases, the results from the two sensors produced the same diagnosis conclusion. However, in some situations the two sensors gave different conclusions. For example, in some instances we realized from the tachometer and current signal data are gives the information then the output before fusion has been shown in Fig. 5, and the Fig. 6 shows the only one type of sensor data (Motor current ) based output, in the both case the results are very highly nonlinear and unstable in that case to take any decision form this information is very critical; but when the hybrid information fusion methods introduces it indicated more better stable and more over liner and positive in nature so in that case the automatically the average performance also increasing as shown in the Fig. 7, & Fig. 8. The comparison with statistical methods and the hybrid methods has been shown in Fig. 9.

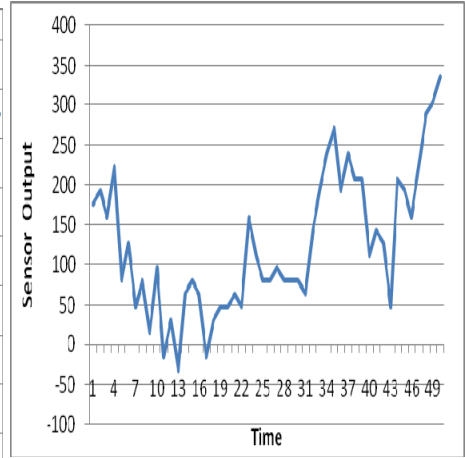




**Fig. 4.** Proposed Multi Sensor Data Fusion Techniques



**Fig. 5.** Sensory signal before Fusion



**Fig. 6.** Only S1 types of Sensor Data Fusion output

It is clear from comparison result in Fig. 9, that the hybrid methods gives better performance than the standard statistical approaches. SVM-based classifiers are claimed to have good generalization properties compared to conventional classifiers, because in training the SVM classifier, the so-called structural misclassification risk is to be minimized, whereas traditional classifiers are usually trained so that the empirical risk is minimized. When a signal falls outside the clusters, it is tagged as a potential motor failure [13].

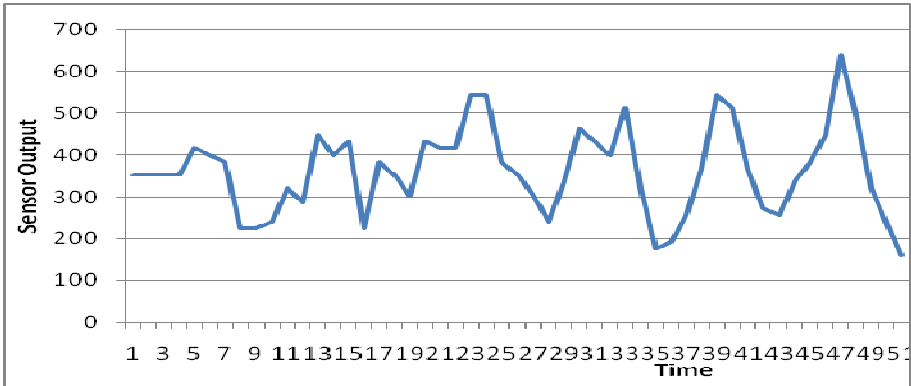


Fig. 7. The Hybrid and Multiple (S1, S2) types of Sensor Data Fusion output

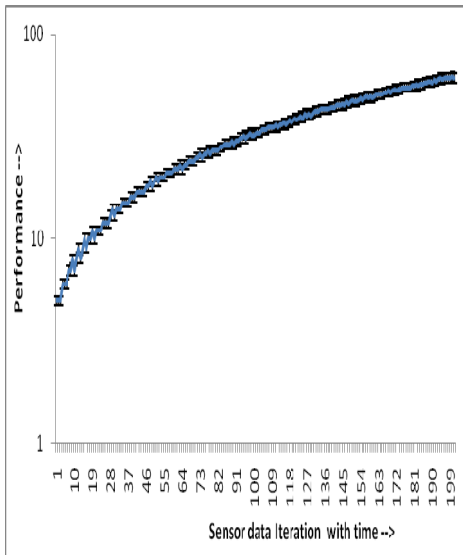


Fig. 8. The performance values with different simulation average taken with standard SVM based classification and Hybrid and multiple type of sensor data fusion

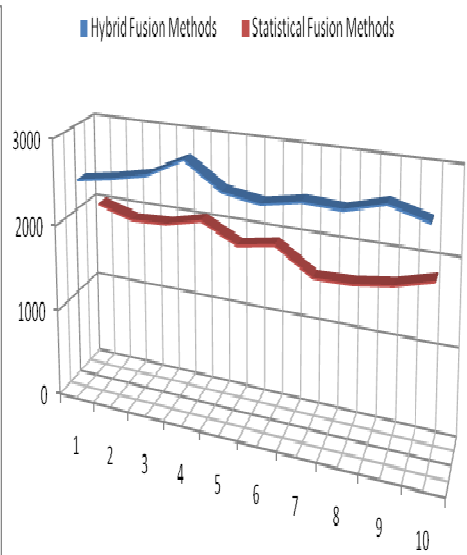


Fig. 9. The comparison with statistical fusion algorithm with hybrid data fusion with iteration

## 6 Conclusion

This paper describes a multi-sensor implementation of an intelligent hybrid BLDC Motor operated valve fault diagnostic system used in aerospace application. The paper discusses the fault diagnostic problem formulated in the context of the theory in terms of fault signal classification. We applied the proposed methods in practical cases. We

presented experimental results to demonstrate how decision conflicts can be resolved and how the performance of hybrid approaches can improve by fusing information from multi-sensors. The experiments give the databases which are done in the laboratory testing. We assumed that the system has potential to serve an intelligent fault diagnosis system in other hard real time system applications also.

## References

- [1] Runkler, T., Sturm, M., Hellendoorn, H.: Model based sensor fusion with fuzzy clustering. In: The 1998 IEEE International Conference on Fuzzy Systems Proceedings, IEEE World Congress on Computational Intelligence, vol. 2, pp. 1377–1382 (May 1998)
- [2] Luo, R., Kay, M.: A tutorial on multisensor integration and fusion. In: 16th Annual Conference of IEEE Industrial Electronics Society, IECON 1990, pp. 707–722 (November 1990)
- [3] Luo, R., Yih, C., Su, K.: Multisensor fusion and integration: approaches, applications, and future research directions. *IEEE Sensors Journal* 2, 107–119 (2002)
- [4] Lee, C., Xu, Y.: Theoretical study on a new multi-sensor system. In: Proceedings of the First ISA/IEEE Conference Sensor for Industry, pp. 187–191 (November 2001)
- [5] Abderahman, M., Kandasamy, P.: Integration of multiple sensor fusion in controller design. In: Proceedings of the American Control Conference, vol. 4, pp. 2609–2614 (May 2002)
- [6] Banerjee, T.P., Das, S., Roychoudhury, J., Abraham, A.: Implementation of a New Hybrid Methodology for Fault Signal Classification Using Short -Time Fourier Transform and Support Vector Machines. *Advances of Soft computing*, pp. 219–225. Springer, Heidelberg, ISBN 978-3-642-13160-8
- [7] Rubezic, V., Djurovic, I., Dakovic, M.: Time-frequency representations based detector of chaos in oscillatory circuits. *Signal Processing* 86(9), 2255–2270 (2006)
- [8] Boashash, B. (ed.): *Time frequency Signal Analysis and Applications*. Elsevier, Amsterdam (2003)
- [9] Vapnik, V.N.: *The nature of statistical learning theory*. Springer, New York (1999)
- [10] Vemuri, A., Polycarpou, M.: On the use of on-line approximations for sensor fault diagnosis. In: Proceedings of the American Control Conference, vol. 5, pp. 2857–2861 (June 1998)
- [11] Durrant-Whyte, H.: Elements of sensor fusion. *IEE Colloquium on Intelligent Control*, 5/1–15/2 (1991)
- [12] Alur, R., Dill, D.: The Theory of Timed Automata. *Theoretical Computer Science* 120, 143–235 (1994)
- [13] Mall, R.: *Real time systems. Theory and practice*. Pearson Publication (2007)
- [14] Boashash, B. (ed.): *Time frequency Signal Analysis and Applications*. Elsevier, Amsterdam (2003)

# Steganalysis for Calibrated and Lower Embedded Uncalibrated Images

Deepa D. Shankar, T. Gireeshkumar, and Hiran V. Nath

TIFAC CORE in Cyber Security  
Amrita Vishwa Vidyapeetham,  
Coimbatore, India

{sudee99,gireeshkumart,hiranvnath}@gmail.com

**Abstract.** The objective of steganalysis is to detect messages hidden in a cover images, such as digital images. The ultimate goal of a steganalyst is to extract and decipher the secret message. In this paper, we present a powerful new blind steganalytic scheme that can reliably detect hidden data with a relatively small embedding rate in JPEG images as well as using a technique known as calibration. This would increase the success rate of steganalysis by detecting data in transform domain. This scheme is feature based in the sense that features that are sensitive to embedding changes are being employed as means of steganalysis. The features are extracted in DCT domain. DCT domain features have extended DCT features and Markovian features merged together in calibration technique to eliminate the drawbacks of both(inter and intra block dependency) respectively . For the lesser embedding rate, the features are considered separately to evolve a better classification rate. The blind steganalytic technique has a broad spectrum of analyzing different embedding techniques The feature set contains 274 features by merging both DCT features and Markovian features. The extracted features are being fed to a classifier which helps to distinguish between a cover and stego image. Support Vector Machine is used as classifier here.

**Keywords:** Steganalysis, DCT, Markov, Calibration, Support Vector Machine.

## 1 Introduction

Steganography is a means of communication in a covert manner so that anyone who inspects the message being exchanged cannot collect enough evidence to prove that the message has data hidden in it. This is accomplished by hiding data within an innocent looking image. Steganography should thus make the communication invisible.

To mount an attack on a steganographic scheme, we need to show that it is possible to detect hidden data with a probability greater than random guessing. In this paper we propose a new steganalytic technique which can be applied to different steganographic schemes and image format. However it can be ideally used in JPEG format. We constrain ourselves to JPEG format since it is most widely used format to be used over the internet.

Steganalysis can be broadly classified as Blind Steganalysis and Targeted Steganalysis. Targeted Steganalysis are designed for a particular steganographic algorithm. This technique is more robust since it has good detection accuracy for that specific technique when they used against a particular steganographic technique. Blind Steganalysis are schemes which are independent of any specific embedding technique are used to alleviate the deficiency of targeted analyzers by removing their dependency on the behavior of individual embedding techniques. Hence one technique can work in a broad spectrum of steganographic techniques. This approach alleviates the deficiency of specific steganalysers by removing their dependency on the behavior of individual embedding techniques. To achieve this, a set of distinguishing statistics that are sensitive to a wide variety of embedding operations are determined and collected. These statistics, taken from both cover and stego images are used to train a classifier, which is subsequently used to distinguish between cover and stego images. Hence, the dependency on a specific embedder is removed at the cost of finding statistics that distinguish between stego and cover images accurately and classification techniques that are able to utilize these statistics.

Universal steganalysis is composed of two important components. These are feature extraction and feature classification. In feature extraction, a set of distinguishing statistics are obtained from a data set of images. There is no well defined approach to obtaining these statistics, but often they are proposed by observing general image features that exhibit strong variation under embedding. The second component, feature classification, operates in two modes. First, the obtained distinguishing statistics from both stego and cover images are used to train a classifier. Second, the trained classifier is used to classify an input image as either being a clean image or carrying a hidden message. Previous literature [1] state only the application of JPEG images in the either DCT domain or in spatial domain [3] in terms of embedding and extraction. This is due to the fact that JPEG images are mostly used in the internet for transmission. Feature based steganalysis [1] [4] is a technique wherein certain features that are sensitive to embedding changes but insensitive to image content is extracted. A set of distinguishing features are obtained from DCT domain. This paper intends to merge both DCT and Markovian features [2] with a possibility of eliminating the drawbacks of both together with calibration technique .But this will not give a proper classification when the embedding rate is as little as 10% .Hence we also use sets of DCT and Markovian features separately to obtain a better classification .The previous literature mentions about the high computational costs of other classifiers like non linear SVM. The features are normalized to increase the computational complexity. In order to reduce further computational complexity or costs, and to obtain reasonable success, SVM is used as a classifier for the DCT domain. Fridrich [1] had developed a blind steganalytic scheme for feature based steganalysis.Fridrich et al [2] used standard 274 features by merging DCT and Markov features for JPEG steganalysis.

In the next section, we will discuss about the general architecture of the system. Section 3 will deal with the implementation issues regarding the architecture. The experimental results are discussed in section 4.Section 5 will have a short note on the future work.

## 2 Implementation Issues

Below is the system architecture of the feature based steganalytic system using DCT .The concept of feature extraction is combined with linear classification to devise an analytic system mainly for JPEG images.. It has been understood in literature survey that calculating the features directly in JPEG domain is more sensitive to a wider type of embedding algorithms. The direct calculation also enables a more straight forward interpretation of the influence of individual features on detection as well as easier formulation of design principles leading to more secure steganography.

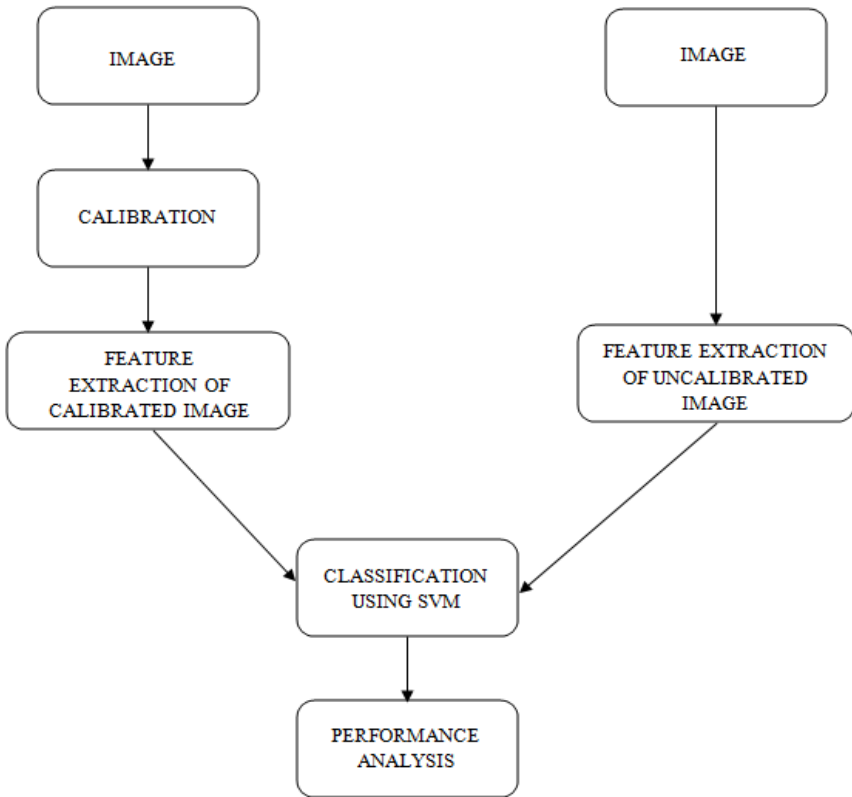


Fig. 1. System Architecture

### 2.1 Feature Extraction

The goal of the paper is to merge new feature set for calibration which gives a better detection rate than any other steganalytic technique and to get a good classification rate for embedding less than or equal to 10 % for uncalibrated image. The proposed feature set is used to construct a general linear classifier. The first step is to extract the

features, and then the features are normalised to improve the algorithm efficiency. Then the Support Vector Machine is designed with respect to accuracy, reliability and cost to give best results. The SVM is trained by the obtained features and then it is subjected to testing on images which were used during training and also on images which are not trained. JPEG format of images are considered here. Figure 1 shows the overall representation of the system.

Four types of features are extracted in DCT domain. They are First Order features, DCT features, Extended DCT features and Markov features. The first order statistics include, mean, standard deviation, skewness and kurtosis of pixels. DCT features are global histogram, individual histogram, dual histogram, variance, blockiness and cooccurrence. The original DCT features [1] have 23 functionals in them. The original DCT features can be extended to extended DCT features [2] which can extract about 193 functionals. Another set of features is the Markov features [2] whose dimensionality can be reduced to 81. While the extended DCT features model inter-block dependencies between DCT coefficients, the Markov features capture intra-block dependency among DCT coefficients of similar spatial frequencies within the same 8X8 block. Hence they have been merged to eliminate the drawbacks of both. Another reason for merging the set is that the classifiers employing each feature set individually have complementary performance. Markov features are unable to detect short message lengths. Hence feature sets are merged giving low false positive rates and high detection accuracy. There is a dip in detectability for 25% message length for JPHide & Seek around the quality factor of 90 which is not accounted. Messages less than or equal to 10% is checked for detectability here. All features in the feature set are uncalibrated.

Calibration is a process through which one can estimate macroscopic properties of cover image from the stego image. Calibration is usually done in DCT domain which gives an estimate of cover image. It will be close to cover image in terms of perceptibility. Calibration is a technique by which a JPEG image J1 is decompressed, converted to spatial domain, four pixels each are cut horizontally and vertically. This image is later converted back to DCT using the same quantization matrix. This is equivalent to shifting of an image, which helps it to retain the DCT coefficients. But the embedded data will be erased making the image a close estimate of the cover image. The newly obtained JPEG image J2 has most macroscopic features similar to the original cover image because the cropped image is visually similar to the original image. The original DCT features were extracted as L1 norm of the absolute value of the difference between the cover image and stego image. This process removes many relevant features needed for analysis. Hence certain functionals with proposed differences were used in DCT. They are called extended DCT features. The Markov feature set as proposed in [2] models the difference between absolute values of neighboring DCT coefficients as a Markov process. The Markovian functionals taken together will comprise of 324 features. This has increased dimensionality, which can be reduced by taking the average of the four 81 dimensionality features. Thus we get a combined set of Extended DCT and Markov features as 274.

## 2.2 SVM Prediction

Support Vector Machine is a supervised learning technique for classification. There are many techniques for classification like neural network, perceptron, Fisher Linear Discriminant and SVM. Out of this, SVM is widely popular in Machine Learning since it maps the data from the original space into a high dimensional feature space. When the kernel function is linear, the resulting SVM is a maximum margin hyperplane. Given a training sample, the hyper plane splits a given training sample in such a way that the distance from the closest cases to the hyper plane is maximized. The complexity of SVM depends on the training samples. Hence we can conclude that SVM guarantees generalization to a great extend.

## 3 Experimentation Result

### 3.1 Database of Images

One of the important aspects of any performance evaluation work is the dataset employed in the experiments. The dataset needs to include a variety of textures, qualities and image formats. A set of 42 images were taken with JPEG and compressed to a size of 256 X 256. We choose a large amount of JPEG format due to its wide popularity in transmission through the internet. A practical evaluation of project is presented by testing unconditional steganalysis for two different algorithms with diverse embedding mechanism: F5 and PVD. Unless stated otherwise, all results were derived on samples from the testing set that were not used in any form during training. The JPEG format is used for study. 42 datasets of cover image and stego image is taken for analysis. These images are used to extract different features like first order features, DCT features, Extended features etc. Many features of these datasets maybe irrelevant. Hence these features maybe removed for better performance. The output is given to a linear SVM. Out of the 42 datasets used, 34 are used to train the SVM. 8 datasets are used for testing.

Since the merged features of uncalibrated images have 274 features, they tend to give low detectability rate. Hence first order features, extended DCT features and Markovian features are taken separately for classification. The steganographic algorithm used here is F5. After the classification, the results obtained is tabulated in table 1.

**Table 1.** 10% Embedding with separated features

PERCENTAGE EMBEDDING		EXTENDED DCT				MARKOV
	MOMENT	GLOBAL HISTOGRAM	VARIANCE	CO-OCCURANCE	BLOCKINESS	
10%	70	60	80	84	100	80



A combined merged DCT and Markov feature resulted in lower classification where as separating features gave a better classification. Moreover even with separated features, it can be concluded that inter-block features like variance,co-occurrence and blockiness give a better classification rate than intra-block features like moment and global histogram.

Another important result is the one with calibration. Here all 274 features of Extended DCT and Markov features have been taken into consideration. Different embedding rates have also been considered for classification.A comparative study has been conducted between images with calibration and images without calibration.The results obtained thus for uncalibrated images is tabulated in table 2.

**Table 2.** Classification of uncalibrated images using SVM

EMBEDDING PERCENTAGE	10	25	50	10-25-50-70
F5	59.5	46.6	60	50
PVD	57.2	54.5	83	50

The results obtained for calibrated images is tabulated in table 3.

**Table 3.** Classification of calibrated images using SVM

EMBEDDING PERCENTAGE	10	25	50	10-25-50-70
F5	83.33	89.3	95	75
PVD	83	91	98	75

From the above two tables, we can conclude about a better classification result with calibrated images rather than uncalibrated images.

### 3.2 Linear SVM

The dimensionality reduction needs to be done because the classifier used here is linear SVM. There are many classifiers like neural network, perceptron, linear fisher discriminant, SVM etc. Out of these, SVM is considered to be more powerful in terms of classification. The feature reduction is mainly due to the use of linear SVM to reduce the cost and computational complexity. Since the steganalysis system used was Blind, the SVM has to be trained before any testing occurs. Out of the 42 images, 34 images were used to train the data and the rest 8 were used to test the data.

## 4 Conclusion and Future Work

A set of features for steganalysis of JPEG and BMP images with a range of quality factors was developed. We considered features that take into account the numerical

changes in DCT coefficient introduced by embedding. The feature set was obtained by merging and modifying two previously proposed feature sets with complementary performance (the DCT features that capture the inter-block dependencies among DCT coefficients and Markov features which capture intra-block dependencies). In particular, we used the DCT features by replacing the L1 norm in their calibration by differences and added calibration to Markov features and reduced their dimensionality. According to the experiments, the new merged feature set provides better results than previous results.

The present feature based system has PCA incorporated for reduced dimensionality thereby obtaining a better feature set. This feature set is being input into the linear SVM for classification between cover and stego image. Apart from the features mentioned, it has been decided to find a set of calibrated set of 274 features and uncalibrated set of 274 features [5]. feature selection can be done using independent component analysis. These features can be later fed to a classifier, probably a soft margin classifier with Gaussian kernel [5]. The classification accuracy of the calibrated features as compared to uncalibrated features can be estimated .

Another enhancement of the paper can be an introduction to estimation of payload. This is called quantification, which can be achieved by means of Support Vector Regression [6].

Apart from the features mentioned, it has been decided to add a new feature set called Binary Similarity Measure. These features are obtained from the spatial domain representation of the image. This is calculated from the seventh and eighth bit planes of an image .The correlation between the contiguous bit planes decrease after a message is embedded in the image. Thus features like histogram, similarities, entropy related features etc are calculated.

The next enhancement is on the detection of message length. The method consists of a sequence of estimation procedures that use spatial domain representation of the cover and stego images to estimate the length of a message embedded.

From the digital signal processing point of view, this type of embedding can be considered as adding a certain type of noise to the cover image obtaining a stego image. Because of the undetectability requirement, stego message embedding is performed with a low stego message/cover image ratio. This method works in three steps. In the first stage, the cover image is estimated from the stego image. Secondly, the stego image is removed from the mixture. The third stage consists of the estimation of stego message length .A potential advantage of this approach is the possibility that not only message length, but also estimates for the location and sign of the embedding changes are obtained. Due to the absence of a priori knowledge about the message length, the Maximum Likelihood is used for estimation of the stego message length. MAP estimator can also be proposed as another approach.

The final enhancement is analyzing the LSB using autocorrelation between pixels. This is done with the assumption that the neighboring pixels are equal in value. The LSB planes matrix is formed. All 0's are replaced by -1 forming a new matrix. Every bit in the matrix is multiplied by itself and the results are all summed. If the two LSB s are equal, their multiplication is 1 else it is -1. So unequal bits decreases the correlation and equal bit increases. After extracting suitable separating features,

weighted Euclidean distance measure are used. There are two initial sets of Stego and non-stego bit planes from which classifiers should be tuned. The centers and variance vectors of each set is computed and their Euclidean distance is found with the two centers. The results are known as auto correlation vector. The incoming vector is classified as stego if the Euclidean distance between the stego and center is less than the Euclidean distance.

## References

1. Fridrich, J.: Feature-Based Steganalysis for JPEG Images and Its Implications for Future Design of Steganographic Schemes. In: Fridrich, J. (ed.) IH 2004. LNCS, vol. 3200, pp. 67–81. Springer, Heidelberg (2004)
2. Pevn'y, T., Fridrich, J.: Merging Markov and DCT Features for Multiclass JPEG Steganalysis. In: Proceedings SPIE, Electronic Imaging, Security, Steganography and Watermarking of Multimedia Contents IX, San Jose, CA, January 29-February 1, vol. 6505, pp. 301–314 (2007)
3. Yadollahpour, A., Niami, H.M.: Attack on LSB Steganography in Color and Grayscale Images using Autocorrelation Coefficients. European Journal of Scientific Research 31(2), 172–183 (2009) ISSN 1450-216 X
4. Kharrazi, M., Sencar, H.T., Memon, N.: Performance study of common image steganography and steganalysis techniques. Journal of Electronic Imaging 15(4), 041104 (2006)
5. Kodosky, J., Fridrich, J.: Calibration Revisited. In: ACM Multimedia and Security Workshop, Princeton, NJ, September 7-8, pp. 63–74 (2009)
6. Pevny, T., Fridrich, J., Ker, A.D.: From Blind to Quantitative Steganalysis. In: SPIE, Electronic Imaging, Media Forensics and Security XI, San Jose, CA, January 18-22, pp. 0C 1–0C 14 (2009)

# An Efficient Feature Extraction Method for Handwritten Character Recognition

Manju Rani and Yogesh Kumar Meena

Malaviya National Institute of Technology, Jaipur, India  
manju\_gautam14@yahoo.com, yogimnit@gmail.com

**Abstract.** Handwritten character recognition in a particular language is one of the favourite topics for research from two last decades. Image processing and pattern recognition plays a lead role in handwritten character recognition. It is not a easy task to build a program to achieve hundred percent accuracy for handwritten characters because even humans too make mistakes to recognize characters. There are three main steps of handwritten character recognition- Data collection and preprocessing, feature extraction and classification. Data collection includes creating a raw file of handwritten character images. Preprocessing steps are applied to find a normalized binary image of handwritten character which is easy to process in next step. Feature extraction is the process of gathering data of different samples so that on the basis of this data we can classify samples with different features. Feature extraction from preprocessed handwritten character plays the most important role in character recognition. Thus feature extraction stage in handwritten character recognition system has a large scope for researchers. In this paper, we also introduce a new feature extraction method for handwritten characters named Cross-corner. We use the results of some promising feature extraction methods to find the best method for this application.

**Keywords:** Preprocessing, Feature Extraction, Recognition rate, Classification.

## 1 Introduction

Handwritten characters have large range of writing style from one writer to another writer. That is why, handwritten character recognition is not a difficult task for humans, but for a machine it is really tough. So because of differences in handwriting, an accurate method has not been developed till now. Recognition accuracy of handwritten character depends on the database that are used for recognition. Unwanted slants, skews curves will make difficult it to recognize a handwritten character. Lot of methods for recognizing handwritten characters been developed, if we take a look at the work done in past. Most of the handwritten character recognition methods used neural network because it is easy to use and gives good classification efficiency. Although the research on handwritten character recognition has been going for a few decades, The target of building

100 % accurate system is still out of reach. Feature extraction step plays an important role in handwritten character recognition system because preprocessing and classification stages are almost similar for characters having large variance. A high accuracy character recognition technique feature extraction techniques is presented in this paper that extracts features based on the two diagonal directional lines within a character image. The remainder of this paper is broken down into 6 sections. Section 2 describes the literature of work done in handwritten character recognition. Section 3 contain the detailed description of feature extraction technique with brief introduction of preprocessing, classification using backpropagation takes place in section 4, and section 5 provides experimental result and analysis. Finally section 6 presents conclusions and future research.

## 2 Related Work

Most basic way for pattern recognition is based on the Bayesian Theory. This statistical approach is called Handwritten Character Recognition Using Bayesian Filter [11]. This method of recognizing patterns uses probability theory for decision making. This technique takes handwritten characters read by a scanner as an input. Experimental results of this method indicated that the method has high recognition performance in spite its simplicity. Handwritten Character Classification Using Nearest Neighbor is based on a simple statistical technique which is used for large database Performance of these systems changes proportionally with the size of the training database [12]. Three distance metrics for the nearest neighbor classification system may be Euclidean distance or Manhattan's distance between pixels of an input vector. Template matching is also a standard oldest technique for handwritten character recognition.

Now we discuss the neural network for handwritten character recognition which is used most commonly because of its benefits like fault tolerance, nonlinearity, adaptivity and supervised learning etc. Thus we select the neural network as a classifier for handwritten character recognition. In this section, we take an overview of the neural network classification procedure according to different authors. According to Velappa Ganapathy a neural network training method is can be used for higher resolution character images. This method [4] can produce accuracies of at least 85% and more. Backpropagation neural network with boosting is used in this paper for classification purpose.

Now according to Xin Wang a handwritten character recognition based on backpropagation algorithm is also proposed. In this method [9] binary matrix of the image is passed to the input of backpropagation network. Momentum and variable learning rate are two additionally added terms in backpropagation. According to him algorithm recognition percentage is 95% on visual studio 2005 platform. A paper Handwritten English character recognition using neural network by Anita Pal and Dayashankar Singh purposed a high accuracy and minimum training time neural network is used for character recognition [6]. Fourier Descriptor method is used for feature extraction. Recognition rate of skeletonization and normalized binary image of character is 94%.

According to the paper by Chongliang Zhong Self Organizing Competition network is used to classify the characters. This paper [7] introduced 13-point feature of skeleton method to extract the data containing peculiar properties of handwritten character. Next paper Diagonal feature extraction based on handwritten character system using neural network by J.Pradeep, presented a diagonal feature extraction method [8] which lead to increase in accuracy rate. Diagonal direction features increased the recognition rate drastically in comparison to horizontal and vertical direction. Now we studied the MATLAB handwritten character recognition by neural network having geometry based features of handwritten characters. This literature [2] explains many techniques for segmented character to enhance the accuracy rate. Features are based on basic line type and structure of the character. A paper referenced [3] by S.V. Rajashekararadhya described the zone and centroid based feature extraction algorithm. This algorithm is more suitable for character having curved lines. Neural network and backpropagation algorithm are used as classifiers for these features to recognize characters. A paper Handwritten Character Recognition Using Twelve Directional Feature Input and Neural Network [10] presented a method based on direction of each pixel. Direction can be found by calculating gradient of the pixel. According to Sandhya Arora and Debotosh Bhattacharjee a hybrid method can be build by combining multiple features of different types [5]. From the above papers we observed that neural network classifier is common in all, only with little modification in network. Output changes mainly because of the feature extraction methods. A paper referenced [1] describes multiple methods for feature extraction. In next section we elaborate our method of feature extraction to overcome the drawbacks of previous methods.

### 3 Proposed Feature Extraction Technique

In this section, we will complete the following tasks. Receiving the handwritten character image, preprocessing and feature extraction with cross corner method. After going through the literature of feature extraction methods we observed the following conclusion. In region and frequency based methods we should know the which area of image is dense and which have only few scattered foreground pixels before dividing into zones. Diagonal feature extraction methods are better than region based methods because information is more precise, but these give good results for only some particular languages. Algorithm of diagonal method is little complex than region based methods. Zoning based methods able to find the local characteristics instead of global characteristics. Image Centroid and zone centroid method works well for only characters having maximum curves. Geometry based methods is specific to a language i.e. it is good for such character formed by straight line and simple curves. Now we see that twelve direction method is very conceptual method which used Sobel's mask. In twelve direction method, each feature value is dependent on other value. This method cannot guarantee the increase in accuracy because feature space is large and each feature value range is 0-11. Now we discuss briefly all steps for handwritten character recognition system.

### 3.1 Image Acquisition and Preprocessing

The images in dataset contains handwritten character images in a format obtained directly from the source which may be scanner or tablet PC. We will create a dataset of 26 uppercase alphabets. Now we should search the method for reducing the data amount and simplifying data. These methods comes under the preprocessing steps. A classifier executes fast and gives better results if the input vectors are preprocessed before feeding the dataset into the network. The important role of preprocessing is to avoid peculiar output results from the classifier. The functioning of preprocessing is mainly divided into 4 steps. First step is binarization, which transform a gray-scale image into binary image. We see that some person write small sized character, some writes large sized character, some writes from the left corner in the given space and some writes in center of the space. So to avoid this problem only the useful part is cropped by drawing a rectangle surrounding the binary image and scaled to a fixed size. That was our second step. After getting a uniform size images, in third step of preprocessing slant removal methods are applied to convert all character images to a standard form. Last step is thinning which deletes the redundant pixel of the image and in parallel it ensure that the contour of the character image do not change.

### 3.2 Cross-Corner Feature Extraction Technique

We know that, the main goal of feature extraction is improve the speed and accuracy of the classifier for pattern recognition. Features of the character are extracted such that whole portion of binary image is covered and each portion have a distinct property. In this section we will introduce the cross-corner feature extraction method. Before finding feature vector we will divide the preprocessed image into zones so that we can calculate the local characteristics instead of global characteristics. This method divides the image into 'n' non-overlapped equal zones as shown in figure 1. If whole image is divided into nine zones, the

1	2	3	4
.	.	.	.
.	.	.	.
n-3	n-2	n-1	n

Fig. 1. Zoning of Image

way of finding features of cross-corner of character 'A' are shown in figure 2. First we divides the binary image into nine zones, then calculate number of right and left slanted line in each zone. After getting individual features of each zone, combine them to make a single feature space of that character.

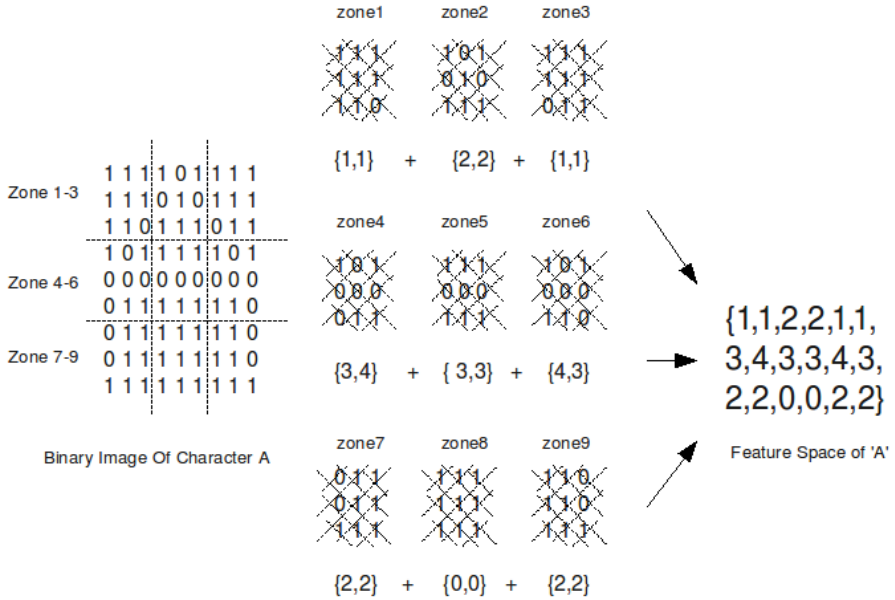


Fig. 2. Feature Space of Character 'A'

The method divides the whole image into 2 directions left slanted and right slanted. So after dividing the image into zones count the number of left diagonal and number of right diagonal lines starting from the top-left and top-right corner respectively . Number of left diagonal line is one feature and number of right diagonal line is another feature in each zone.

### 4 Classification Method

Application of neural networks for handwritten character recognition is very common and important. Handwritten character recognition can be implemented by using a backpropagation neural network(BPN) that has been trained according to train dataset. During training, the network is trained to associate output characters with input characters. Trained network is used to identify the associated output character of input characters. Backpropagation network consists of three layers-input, output and hidden layer. Each layer contain a fixed number of neurons. Dimension of input layer depends on the number of features of handwritten character sample. In case of cross corner it is 50, if number of zones are 25. The number of output layer neurons depends on the number of possible classes in training dataset. So, number of neurons in output layer are 26(26 uppercase English letters). Number of hidden neurons are arbitrary. Weight initialization and selection of activation function depends on the application.



## 5 Results and Analysis

**Datasets Description:** Total number of samples are 650 in both dataset1 and dataset2. BMP images written by 25 writers of each capital English alphabet(A-Z) are taken. First 15 characters are less noisy and next 10 characters are more noisy in dataset1. In dataset2, sequence of noisy characters is random. Dataset1 is taken from MATLAB, which is prepared by Dileep Gaurav. Both datasets are divided into 4 training sets and 4 test sets according to table 1. After dividing the datasets we apply techniques for feature extraction for handwritten characters, likewise a comparison between different methods.

**Table 1.** Trainset and Testset Distribution

Name	Training	Testing
TD1	1-15	16-25
TD2	11-25	1-10
TD3	1-20	21-25
TD4	6-25	1-5

Simulation results of dataset1 and dataset2 on BPN are shown in table 2 and 3 respectively. First column contains the name of feature extraction method name, Second column shows the number of hidden layer, third, fourth, fifth column contain results of TD1, TD2, TD3 and TD4 respectively. Each method runs for 2000 epochs in BPN. Weights are random between -1 to 1. Mean square error function is used in BPN.

We know that, Capital English characters includes simple horizontal, vertical, left diagonal, right diagonal lines and simple curves. Curves role is in only some characters. The method divides the whole image into 2 directions left slanted and right slanted. With only these directions it tells that why our method works well. For example this method can differentiate between 'X' and 'Y', where diagonal method cannot.

After Observing results of dataset1 and dataset2 on BPN and feature vectors, now we can compare these methods in terms of efficiency and which is suitable for which type characters. 16-region and 25-region methods based on pixel-frequency of a particular zone, gives good accuracy but zoning need complete understanding of characters in the dataset. Information loss is high in 16-region in comparison to 25-region. Diagonal and vertical line methods are based on type of lines of foreground pixels instead of frequency of foreground pixel as in 16-region and 25-region methods. These methods give more exact information and these are more specific to characters build by straight lines. Diagonal method is better than vertical method because it provide more fine information about a particular zone a character. Centroid based method depends on the distance of foreground pixel from the centroid of character image and centroid of the each zone of that character. That is why centroid based method work good for those

**Table 2.** Recognition Rate on Dataset1

Feature Extraction Method	Feature space size	Number of Hidden Nodes	TD1	TD2	TD3	TD4
16region	16	20	16	52	26	42
25region	25	30	70	81	78	92
Diagonal Line	25	30	68	90	76	90
Vertical Line	25	30	60	76	76	83
Geometry-Based	54	42	56	70	63	65
Centroid-Based	50	40	70	81	78	94
Direction-Based	100	50	68	90	76	94
<b>Cross-corner</b>	50	42	70	93	83	97

**Table 3.** Recognition Rate on Dataset2

Feature Extraction Method	Feature space size	Number of Hidden Nodes	TD1	TD2	TD3	TD4
16region	16	20	62	60	57	782
25region	25	30	45	58	60	63
Diagonal Line	25	30	75	83	81	86
Vertical Line	25	30	48	60	79	65
Geometry-Based	54	42	45	59	70	60
Centroid-Based	50	40	50	64	81	70
Direction-Based	100	50	50	70	85	86
<b>Cross-corner</b>	50	42	76	80	82	93

characters having maximum curves. Geometry based method find the type of lines (horizontal, vertical, left diagonal and right diagonal) with length of lines after zoning. Geometry based method gives is more accurate features but feature space is large. Twelve-direction based method does not require zoning. Feature space is as large as there is grid size of character image. Each feature value have a direction based on its 8 neighbors. Accuracy is optimal but feature values are redundant. At last, our proposed method– Cross-corner provide substantial increase in accuracy without increasing much in feature space size.

## 6 Conclusion

This paper presents a new feature extraction technique for handwritten character recognition. The feature extraction technique(cross-corner) compared to another popular technique in the literature using Backpropagation Neural Network classification scheme. It outperforms as compared to the other feature extraction techniques. In future a number of considerations will be addressed including an improved preprocessing methodology, a more advanced approach to character image generation, an investigation of a wide variety of global and local features.

## References

1. Heutte, L., Paquet, T., Moreau, J.V., Lecourtier, Y., Olivier, C.: A structural/statistical feature based vector for handwritten character recognition. *Pattern Recognition Letters* 19, 629–641 (1998)
2. Blumenstien, M., Verma, B., Basli, H.: A novel feature extraction technique for the recognition of segmented handwritten characters. In: *Proceeding of Seventh International Conference on Document Analysis and Recognition* (2003)
3. Rajashekararadhya, S.V., Vanaja Ranjan, P.: Efficient zone based feature extraction algorithm for handwritten numeral recognition of four popular south indian scripts. *Journal of Theoretical and Applied Information Technology*, 1171–1180 (2005-2008)
4. Ganapathy, V., Liew, K.L.: Handwritten character recognition using mutiscale neural network training technique. *World Acedamy of Science, Engineering and Technology*, 32–37 (2008)
5. Arora, S., Bhattacharjee, D., Nasipuri, M., Basu, D.K., Kundu, M.: Combining the multiple feature extraction techniques for handwritten devnagari character recognition. In: *IEEE Region 10 Colloquium and the Third ICIIS, Kharagpur, India*, pages 110 (2008)
6. Pal, A., Singh, D.: Handwritten English Character Recognition using Neural Network. *International Journal of Computer Science & Communications*, 141–144 (2010)
7. Zhong, C., Ding, Y., Fu, J.: Handwritten character recognition based on 13-point feature of skeleton and self-organizing competition network. In: *International Conference on Intelligent Computation Technology and Automation*, pp. 414–417 (2010)
8. Pradeep, J., Srinivasan, E., Himavathi, S.: Diagonal feature extraction based on handwritten character using neural network. *International Journal of Computer Applications* 8, 17–21 (2010)
9. Wang, X., Huang, T.-L., yu Liu, X.: Handwritten character recognition based on BP neural network. In: *Third International Conference on Genetic and Evolutionary Computing*, pp. 520–524 (2009)
10. Singh, D., Singh, S.K., Dutta, M.: Handwritten Character Recognition Using Twelve Directional Feature Input and Neural Network. *International Journal of Computer Applications* 1, 82–85 (2010)
11. Araki, N., Okuzaki, M., Konishi, Y., Ishigaki, H.: A Statistical Approach for Handwritten Character Recognition Using Bayesian Filter. In: *3rd International Conference on Innovative Computing Information and Control*, pp. 194–198 (2008)
12. Smith, S.J., Bourgoin, M.O., Sims, K., Voorhees, H.L.: Handwritten character classification using nearest neighbor in large databases. *IEEE Pattern Analysis and Machine Intelligence* 16, 915–919 (2002)

# Optimized Neuro PI Based Speed Control of Sensorless Induction Motor

R. Arulmozhiyal<sup>1</sup>, C. Deepa<sup>2</sup>, and Kaliyaperumal Baskaran<sup>3</sup>

<sup>1</sup> Sona college of Technology/Department of EEE, Salem, Tamil Nadu, India  
arulmozhiyal@gmail.com

<sup>2</sup> NPR College of Engineering and Technology/Department of EEE,  
Dindigul, Tamil Nadu, India  
deepasekaran@yahoo.co.in

<sup>3</sup> Government College of Technology/Department of CSE, Coimbatore, Tamil Nadu, India

**Abstract.** In this paper a sensorless vector control system of induction motor using Neural Networks is presented. Neural network is used to control the non linear dynamic systems to get desired degree of accuracy. A feed forward neural network with one input, two units in the hidden layer and one output is used for the speed controller. The tracking of the rotor speed is done by a neural PI controller and is realized by adjusting the new weights of the network depending on the difference between the actual speed and the command speed. The use of the controller tracks the rotor speed command smoothly and rapidly, without overshoot and with zero steady state error without the sensor. GA has been recognized as an effective and efficient technique to solve optimization problems. Finally this controller can be optimized using a Genetic Algorithm Technique. When compared to Neuro PI controller Genetic Algorithm produces better performance. Computer simulation results are carried out with various tool boxes in MATLAB to verify the effectiveness of the proposed controller. The result concludes that the efficiency and reliability of the proposed speed controller is good.

**Keywords:** Sensorless vector control, Genetic Algorithm, Neural Network, Backpropagation Network.

## 1 Introduction

Vector control is a control strategy to decouple flux and torque from an induction motor in order to emulate a DC motor. The great advantage is that it can be controlled as easy as a DC motor and induction one with all of its advantages such as high efficiency, robustness, no maintenance and low cost. By using vector representation, it is possible to present the variables in an arbitrary coordinate system. If the coordinate system rotates together with a flux space vector, then we use different terminology: flux-oriented control. In this way, it was possible to represent the electromagnetic torque as a product of flux-producing current and a torque-producing current[1].

Separately excited dc drives are simpler in control because independent control of flux and torque can be brought about [2]. In contrast, induction motors involve a coordinated control of stator current magnitude and the phase, making it a complex control. The rotor flux linkages can be resolved along any frame of reference. This requires the position of the flux linkages at every instant. Then the control of the ac machine is very similar to that of separately excited dc motor at constant flux. Such method will be selected to control our machine.

The new and real applications on the induction motor need more rapid and smooth without overshoot command speed track. PI controller is unquestionable the most commonly used control algorithm the process control industry. The main reason is relatively simple structure, which can be easily understood and implemented in practice[3,4].

In recent years neural networks has gained a wide attention in control applications. The NN provides non-linear modeling of motor drive system without any knowledge of predetermined model and thus makes the system robust to noise, parameter variations, load changes[5,6]. Neural Network can be used to control the non-linear dynamic systems since they can approximate a wide range of non-linear functions to any desired degree of accuracy. The NN could provide one or more system quantities even in the absence of a analytical expression between the inputs and outputs[7]. They have the advantage of that they can be implemented in parallel, which gives relatively fast computation[8].

GA is a stochastic global adaptive search optimization technique based on the mechanisms of natural selection. Recently, GA has been recognized as an effective and efficient technique to solve optimization problems[9,10]. Finally this controller can be optimized using a Genetic Algorithm Technique. When compared to Neuro PI controller Genetic Algorithm produces better performance.

## 2 Sensorless Vector Control

An incremental speed signal for an induction motor is essential for closed-loop speed control of scalar or vector drives. The signal is also needed for indirect vector control, and direct vector control if speed control is necessary from zero speed. A physical speed encoder (typically of the optical type) mounted on the shaft adds cost and reliability problems to the drive, in addition to the need for a shaft extension for mounting it. In modern speed sensorless vector control, precision speed estimation from the machine terminal voltages and currents with the help of DSP is an important topic of research.

Machine Models are used to estimate the motor shaft speed, and, in high-performance drives with field oriented control, to identify the time-varying angular position of the flux vector. In addition, the magnitude of the flux vector is estimated for field control. Different machine models are employed for this purpose, depending on the problem at hand. A machine model is implemented in the controlling microprocessor by solving the differential equations of the machine in real-time, while using measured signals from the drive system as the forcing functions.

The accuracy of a model depends on the degree of coincidence that can be obtained between the model and the modeled system. Coincidence should prevail both in terms of structures and parameters. While the existing analysis methods permit establishing appropriate model structures for induction machines, the

parameters of such model are not always in good agreement with the corresponding machine data. Parameters may significantly change with temperature, or with the operating point of the machine. On the other hand, the sensitivity of a model to parameter mismatch may differ, depending on the respective parameter, and the particular variable that is estimated by the model. Differential equations and signal flow graphs are used in to represent the dynamics of an induction motor and its various models used for state estimation[11].

The machine stationary frame (ds-qs) equations,, contain speed ( $\omega_r$ ) as a variable that can be solved from the known values of  $v_{ds}$ ,  $v_{qs}$ ,  $i_{ds}$ ,  $i_{qs}$  of an operating machine. The simplified forms of equations that are actually solved in real time for speed estimation. Equations (1) and (2) essentially relate to voltage model rotor flux vector estimation, where  $s = 1 - Lm2/Lr Ls$  and  $S = d/dt$ . These equations are derived from the stator equations and express the rotor fluxes in terms of stator voltages and currents. frequency  $\omega_e$  is eliminated and expressed in terms of rotor fluxes and their derivatives.

$$\frac{d}{dt} (\psi_{dr}^s) = \frac{L_r}{L_m} [v_{ds}^s - (R_s + \sigma L_s s) i_{ds}^s] \tag{1}$$

$$\frac{d}{dt} (\psi_{qr}^s) = \frac{L_r}{L_m} [v_{qs}^s - (R_s + \sigma L_s s) i_{qs}^s] \tag{2}$$

Equations (1) and (2) generate the rotor fluxes, which are then substituted to solve the speed. Obviously, the model is complex and highly parameter dependent. Therefore, the accuracy of estimation is expected to be poor, particularly at low speed.

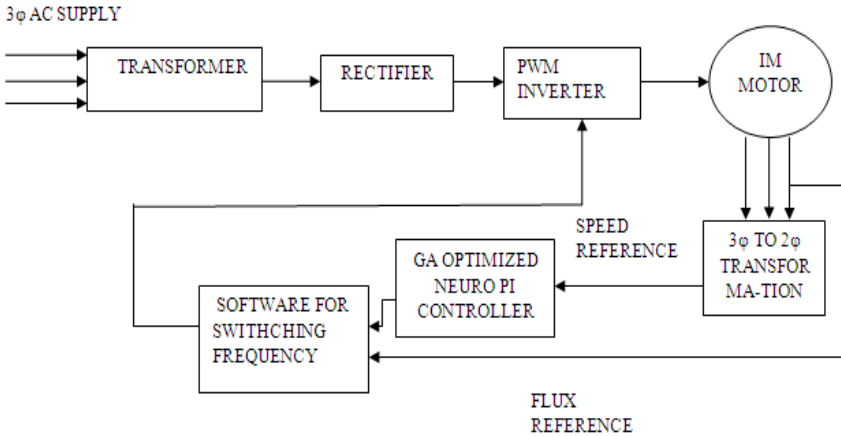


Fig. 1. Basic Block Diagram of the Proposed Scheme

### 3 Neural Network Controller

Since PI controller parameters are to be tuned and it is replaced with the neural controller. Among all the AI techniques, artificial neural network (ANN) or neural network (NNW) is the most important discipline, and its potential impact on power electronics area is tremendous. The technology has a long history, but its development

was camouflaged by the glamorous evolution of modern digital computers. From the early nineties, the momentum of its R & D and applications has surged dramatically. Figure 1 is the basic block diagram of the model in which flux reference and speed reference are taken and simulated.

The optimum number of neurons for hidden layer is chosen as per the table I. Fig 2 shows the simulink block diagram for neural controller.

### 3.1 Back Propagation Algorithm

The type of NNW most commonly used in power electronics is the feedforward multilayer back propagation (backprop or BP) network. The term back propagation comes from the method of supervised training used for the NNW shown. The network is commonly called a multi-layer perceptron (MLP), although the TF can be different from the threshold function. The NNW here has three input signals ( $X1$ ,  $X2$ , and  $X3$ ) and two output signals ( $Y1$  and  $Y2$ ). The circles represent the neurons that have associated TFs and the weights are indicated by dots. The network here has three layers: an input layer, hidden layer, and output layer. With five neurons in the hidden layer the NNW is defined as a 3-5-2 network. The input layer shown is nothing but the nodes that distribute signals to the middle layer. Therefore, this topology is often called a two-layer network. If the signals are bipolar, the hidden layer neurons usually have a hyperbolic tan TF and the output layer has a bipolar linear TF. On the other hand, for unipolar signals, these TFs can be sigmoidal and unipolar linear, respectively. Occasionally, the output layer has a nonlinear TF also. The signals within the NNW are processed in a per-unit manner. Therefore, there is input scaling or normalization and output descaling or denormalization. A constant bias source normally links all of the neurons through weights, the bias connection is not shown for the output layer. The network output signals can be continuous or clamped to 0, 1 or  $-1$ ,  $+1$  levels. Theoretically, a 3-layer NNW is capable of approximating any function with any desired degree of accuracy (universal function approximation), but practically, often more than one hidden layer is used.

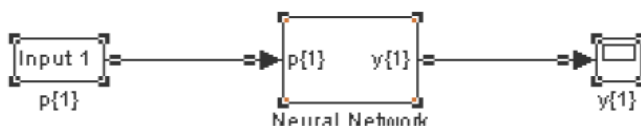


Fig. 2. SIMULINK block for neural controller

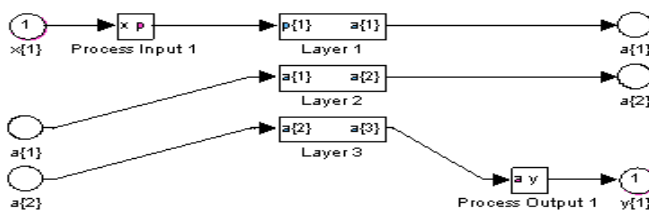


Fig. 3. SIMULINK model of neural network

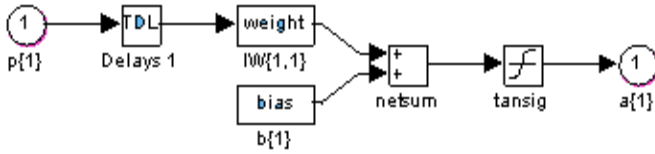


Fig. 4. SIMULINK model of input layer

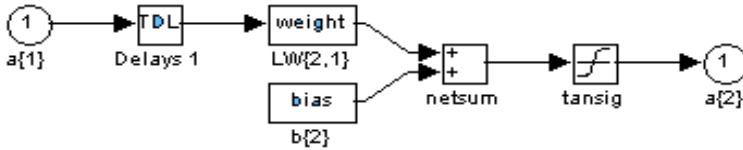


Fig. 5. SIMULINK model of hidden layer

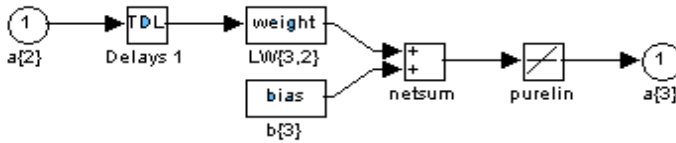


Fig. 6. SIMULINK model of output layer

Table 1. Choice of tansig activation function for hidden layer

Number of hidden layers	Number of epochs with tansig activation function
1	Performance not met
2	Performance not met
3	≈800
4	≈850
5	≈800
6	≈700
7	≈800
8	≈850
9	≈750

## 4 Genetic Algorithm

GAs work with a population of solutions called chromosomes. The fitness of each chromosome is determined by evaluating it against an objective function. The chromosomes then exchange information through crossover or mutation. This



methodology is a derivative free optimization technique based on the concept of natural selection and evolution processes. The basic element processed by a GA is a string formed by concatenating substrings, each of which is a binary coding of a parameter. Each string represents a point in the search space.

GA is a stochastic global adaptive search optimization technique based on the mechanisms of natural selection. Recently, GA has been recognized as an effective and efficient technique to solve optimization problems. Compared with other optimization techniques, such as simulating annealing and random search method techniques, GA is superior in avoiding local minima which is a common aspect of nonlinear systems. Furthermore, GA is a derivative-free optimization technique which makes it more attractive for applications that involve non smooth or noisy signals.

GA starts with an initial population containing a number of chromosomes where each one represents a solution of the problem which performance is evaluated by a fitness function. Basically, GA consists of three main stages: Selection, Crossover and Mutation. The application of these three basic operations allows the creation of new individuals which may be better than their parents. This algorithm is repeated for many generations and finally stops when reaching individuals that represent the optimum solution to the problem.

## 5 PI Controller Design

The PID controller with settings proportional gain ( $K_p$ ), integral time ( $T_i$ ) and derivative time ( $T_d$ ) obtained by using Zeigler – Nichols tuning. The PID controller optimal setting values ( $K_p$ ,  $T_i$  and  $T_d$ ) for proposed system are obtained by finding the minimum values of integral of square of error (ISE), integral of time of square of error (ITAE) and integral of absolute of error (IAE), which is listed in Table 2.

**Table 2.** Simulated Results of Minimum Values of ISE, IAE, ITAE and Optimal setting Values of  $K_p$ ,  $T_i$

ISE	IAE	ITAE	$K_p$	$T_i$ (s)
4.377	0.31935	0.00441557	1.0	0.01

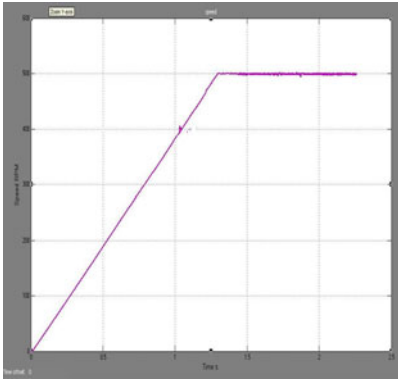
## 6 Result Analysis

The simulation model and results of the proposed Neural controller is evaluated in MATLAB and the speed control of Sensorless Induction Motor is carried out effectively is shown in Fig.7, Fig. 8 and Fig.9. Learning occurs with the learning rate of 0.01 and momentum factor of 0.9. The hyperbolic tangent sigmoid is used as the activation function. Trials have been carried out to obtain maximum accuracy with minimum number of neurons per layer. Back propogation is mainly used in this network for speed control as it is more efficient than any other algorithms and it can be used for non linear applications also.

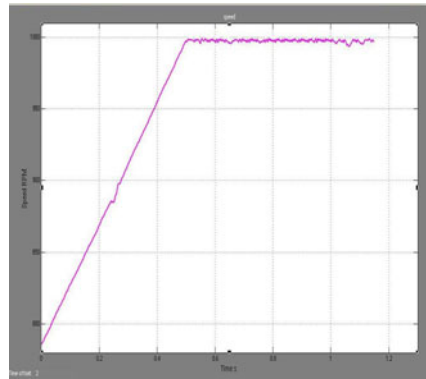
### 6.1 Simulated Result

The Settling time for Fig. 7 is found to be 1.2sec with GA optimized Neuro PI controller for the speed of 500 rpm.

In Fig.8 the settling time for reference speed of 1000rpm (with GA optimized Neuro PI controller) is found to be 0.5 sec.

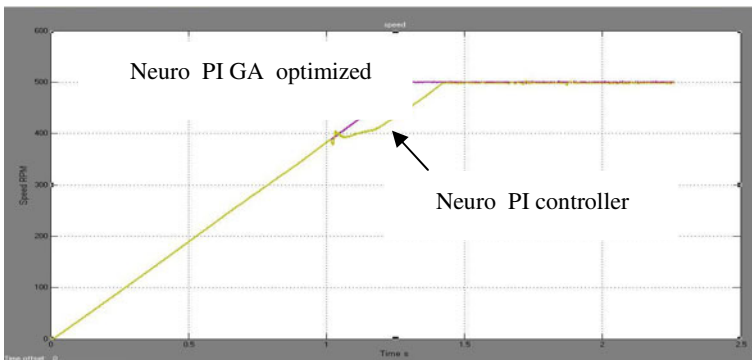


**Fig. 7.** Simulated Response for the reference speed of 500 rpm(with GA optimized Neuro PI controller)



**Fig. 8.** Simulated Response for the reference speed of 1000 rpm(with GA optimized Neuro PI controller)

Finally in Fig.9 the settling time for the reference speed of 500rpm with and without controller is compared and it is found to be less for Neuro controller with GA optimized PI compared to Neuro controller without PI which is 1.4 sec.



**Fig. 9.** Simulated Response for the reference speed of 500 rpm(with GA optimized Neuro PI controller and without controller) comparison

## 7 Training Algorithm

```
P = [-200 -150 -100 -75 -50 -10 -5 -2 -1.9 -1 0 1 2 2.1 5 10 50 75 100 150 200];  
T = [-1 -1 -1 -1 -1 -1 -1 -1 0 0 0 0 1 1 1 1 1 1 1];  
net=newff(minmax(P),[60,10 ,1],{'tansig','tansig','purelin'},'trainscg');  
net.trainParam.show = 50;  
net.trainParam.lr = 0.05;  
net.trainParam.mc = 0.9;  
net.trainParam.epochs = 5000;  
net.trainParam.goal = 1e-5;  
[net]=train(net,P,T);  
a = round(sim(net,P))  
% gensim(net,-1)
```

## 8 Conclusion

In this paper an NNW controller for sensorless control of induction motor was used. The results obtained from the simulation show that the NNW and PI controller has significantly better performance compared to conventional PI controller. The controller has simple form and could be easily designed. The use of this controller caused that the actual speed could track the command rapidly, smoothly and with zero steady state error for the control system without speed sensor. The simulation of the control system with NNW controller is carried out using various tools in MATLAB/Simulink and the results are analyzed. With results obtained from simulation, it is clear that for the same operation condition the induction motor speed control using Neuro controller had better performance than the conventional PI controller, mainly when the motor was working at lower speeds. In addition, the motor speed to be constant when the load varies. Finally this controller can be optimized using a Genetic Algorithm Technique. When compared to Neuro PI controller Genetic Algorithm produces better performance. Real time work can be carried out easily and also Adaptive Neuro Fuzzy Inference System (ANFIS) based Induction Motor speed control can be implemented as a future work by combining both Neural PI and Fuzzy PI controller into a single controller.

## References

- [1] Abu-Rub, H., Hashlamoun, W.: A comprehensive analysis and comparative study of several sensorless control system of asynchronous motor. Accepted to ETEP Journal (European Transaction on Electrical Power) 11(3) (May/June 2001)
- [2] Abu-Rub, H., Awwad, A.K., Motan, N.: Artificial Intelligence Sensorless Control of Induction Motor. IEEE Transactions on Energy Conservation 12(2) (2007)
- [3] Awwad, A., Abu-Rub, H., Guzinski, J., Wlas, M., Krzeminski, Z.: Artificial neural network based sensorless control of induction motor. In: XVIII Symposium Electromagnetic Phenomena in Nonlinear Circuits, Poznan, Poland, June 28-30 (2004)

- [4] Arulmozhiyal, R., Baskaran, K.: Implementation of Fuzzy PI Controller for Speed Control of Induction Motor Using FPGA. *Journal of Power Electronics* 10(1), 65–71 (2010)
- [5] Batran, A., Abu-Rub, H., Guzinski, J., Krzeminski, Z.: Fuzzy logic based sensorless control of induction motors. In: XVIII Symposium Electromagnetic Phenomena in Non Linear Circuits, Poznan, Poland, June 28-30 (2004)
- [6] Ben-Brahim, L., Kudor, T.: Implementation of an induction motor speed estimator using neural networks. In: Proc. IPEC, pp. 52–57 (1995)
- [7] Bose, B.K.: Artificial Neural Network Applications in Power Electronics. In: IEEE Conference on Industrial Electronics Society, pp. 1631–1638 (2001)
- [8] Coello Coello, C.A., Christiansen, A.D.: An Approach to Multi objective Optimization using Genetic Algorithms. In: Intelligent Engineering Systems Through Artificial Neural Networks, vol. 5, pp. 411–416. ASME Press, St. Louis (2000)
- [9] Goldberg, D.E.: Genetic Algorithm in search Optimization and Machine learning. Pearson Education (1986)
- [10] Krzeminski, Z.: Sensorless control of induction motor based on new observer. In: International Conference on Intelligent Motion and Power Conversion, PCIM 2000. Nuremberg (2000)
- [11] Wlas, M., Krzeminski, Z., Guzinski, J., Abu-Rub, H., Toliyat, H.A.: Artificial-Neural-Network-Based Sensorless Nonlinear Control of Induction Motors. *IEEE Transactions on Energy Conversion* 20(3) (September 2005)

# Wavelet Based Fuzzy Inference System for Simultaneous Identification and Quantitation of Volatile Organic Compounds Using SAW Sensor Transients

Prashant Singh and R.D.S. Yadava

Department of Physics, Faculty of Science, Banaras Hindu University,  
Varanasi 221005, India  
{p8singh, ardius}@gmail.com

**Abstract.** Calibrated identification of volatile organics by electronic sensors needs development of data collection and data processing methods that can efficiently generate vapor identity features and some quantitative measure of its concentration simultaneously. In this paper, we present a simulation study on this based on surface acoustic wave (SAW) chemical sensors functionalized by polymer coating. The analysis utilizes transient responses of SAW sensors exposed to seven volatile organic compounds at various concentrations. The feature extraction is done by discrete wavelet decomposition using *Daubechies-2* basis. A fuzzy *c*-means clustering method based Sugeno-type fuzzy inference system was then roped in for simultaneous identification and concentration estimation. The performance of the method has been analyzed for various conditions of polymer film thickness. It is concluded that there exists an optimum region for film thickness over which the present method yields nearly 100% correct classification with less than 1% concentration error.

**Keywords:** Wavelet decomposition, SAW sensor transients, fuzzy clustering and inference, quantitative odor recognition, electronic nose.

## 1 Introduction

Electronic nose is a bioinspired engineering product that is designed on paradigm of mammalian olfactory system. It takes various forms depending on the sensor technology and signal processing methods used. The chemical information about airborne odorants (mostly low molecular weight volatile organic compounds) is generated by chemical selective sensors similar to olfactory receptor neurons in biological nose. Information retrieval from sensor outputs is done by using machine intelligence data processing techniques [1-3]. There are several sensor technologies which have been used for making electronic noses [4-6]. Most common of these are metal-oxide semiconductor (MOS) chemiresistors, conducting polymer composite (CPC) chemiresistors, and quartz crystal microbalance (QCM) and surface acoustic wave (SAW) gravimetric sensors.

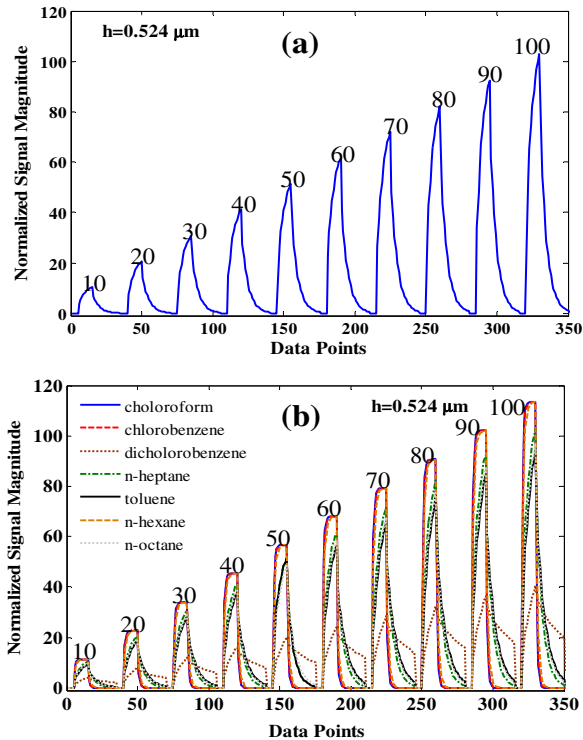
In this paper, we present a simulation study on the simultaneous quantification and identification of volatile organic compounds by using transient responses of a polymer coated SAW sensor. The motivation is to explore a suitable data processing method that can harness information richness of sensor transients for making an efficient SAW vapor detection system. We used discrete wavelet decomposition for transient representation, and fuzzy c-means clustering based fuzzy inference system (FCM-FIS) for quantitation and classification. The simulation data were generated by using a theoretical model of polyisobutylene (PIB)-coated SAW oscillator sensor [7]. The transient sensor responses for exposure to 7 volatile organics: chloroform, chlorobenzene, o-dichlorobenzene, n-heptane, toluene, n-hexane and n-octane were analyzed. In order to represent noisy challenges of real sensing conditions an additive noise source has also been included in the model calculations [7, 8].

## 2 SAW Sensor Model for Transient Response Generation

As mentioned above, we consider a polyisobutylene (PIB) coated SAW delay line oscillator sensor on STX-quartz substrate operating at nominal frequency  $f_0 = 100$  MHz. The vapor exposure and purging is considered to be step concentration function. The moment vapor comes in contact, SAW oscillator frequency starts changing due to partitioning and diffusion of vapor molecules into the polymer coating. The relative change in frequency  $\Delta f/f$  is defined as the signal. The signal rises continuously until steady state condition is reached according to thermodynamic partition coefficient  $K$  defined through  $C_p = KC_v$  where  $C_p$  and  $C_v$  are vapor concentration in polymer and gas phased respectively. The purging occurs when the vapor concentration in gaseous phase is set to zero and desorption is allowed to continue in an inert gas ambient. The theoretical model for SAW delay line oscillator sensor response is developed by Martin, Fry and Senturia in [9]. This has been experimentally validated and used in numerous subsequent publications; see for example [10, 11] and references there in. The theory of transient signal generation under dynamic vapor loading/unloading conditions is presented in detail in [7, 8]. It is based on the theory of polymer coated SAW sensor response [9] and the theory of vapor in- and out-diffusion for uniform homogeneous polymer film on impermeable substrate [12]. In some recent studies the authors have analyzed various aspects of the SAW sensor transient signal generation and vapor recognition [7, 8, 13, 14]. For the present analysis, the transient data for 7 volatile organic compounds (VOCs) for concentration variations over [10–300] ppm were generated. The 5 sec sensing and 10 sec purge durations were employed to generate data at 0.5 sec interval. The sensor outputs were corrupted by an additive noise equivalent to SAW velocity random fluctuations over [-0.0063–0.0063] m.s<sup>-1</sup>.

The model calculation used the following PIB parameters: mass density  $\rho_p = 0.918$  g cm<sup>-3</sup>, complex shear modulus  $G = (1.58 + j0.316) \times 10^9$  dyne cm<sup>-2</sup>, complex bulk modulus  $K = (1 + j0.0) \times 10^{10}$  dyne cm<sup>-2</sup>. The 7 vapors, and their partition coefficient ( $K$ ) and diffusion coefficient ( $D$ ) in PIB for temperatures over (298–325) K written as  $(K, D \times 10^{-11})$  cm<sup>2</sup>s<sup>-1</sup> are: chloroform (200, 260),

chlorobenzene (4680, 230), o-dichlorobenzene (22500, 5.49), n-heptane (1200, 48), toluene (1000, 35), n-hexane (180, 160), n-octane (955, 38). Fig 1 shows the transient response cycles for vapor concentrations varied over [10, 100] ppm at step of 10 ppm and data points at step of 0.5 sec. The polymer thickness for this figure equals to  $h=0.524 \mu\text{m}$  which is equivalent to the phase shift of shear waves radiated into polymer in surface normal direction  $\phi_3/\pi=0.24$  [8]. Fig 1(a) shows single vapor response for n-heptane, and Fig 1(b) shows responses for all the 7 vapors.



**Fig. 1.** Normalized transient response of PIB-coated 100 MHz SAW sensor for vapor concentrations over [10-100] ppm. The polymer thickness is  $h=524 \text{ nm}$ ; (a) responses for n-heptane, and (b) responses for all the 7-vapors as indicated.

The transient data were further transformed by discrete wavelet decomposition (DWT) by using *Daubechies-2 (db2)* basis function [15, 16]. The DWT yields two sets of coefficients – approximation and detail. The coefficients of detail represent noisy components, hence thrown away at successive decomposition stages. The approximation coefficients represent the shape of transients, and are treated as features for data analysis. The DWT results in reduction of data size. The reduction depends on the level up to which decomposition is done. We used here decomposition up to second level. Therefore original data for each vapor class is  $30 \times 35$  matrix (30 concentrations  $\times$  35 data points). After wavelet decomposition data matrix is reduced to  $30 \times 11$  (30 concentrations  $\times$  11 approximation coefficients). This process

produces data denoising as well as variable reduction. The transformed data in feature space (hyperspace defined by wavelet approximation coefficients) were used for vapor identification and quantification as explained below.

### 3 Fuzzy Inference System

The aim of present analysis is to extract vapor identity and concentration information contained in SAW sensor transients for simultaneous identification and concentration estimation of volatile organics. Another objective is to analyze dependence of the results on the sensor parameters (particularly on the polymer thickness) so that the sensor design for optimum performance could be predicted on the basis of simulation results. This is expected to reduce time and cost for sensor development. As an effort in this direction we applied the Sugeno-type fuzzy inference system available in Matlab. We selected '*genfis3*' function with arguments set for fuzzy *c*-means (FCM) clustering based fuzzy inference system (FIS). The *c*-means clustering needs the numbers of clusters to be specified a priori as an input parameter. That is, the data space is assumed to have a fixed number of partitions (clusters or classes). This is possible with supervised learning methods. In the present simulation experiments we are using SAW sensor model generated data for a known number of classes (7 VOCs here). As a first step, it is also important to assess the suitability of the wavelet representation of sensor transients for building fuzzy inference system. For this, we need to experiment with data having known number and type of classes or clusters. This is the second reason for choosing *c*-means clustering. The *c*-means clustering determines the positions of cluster centers in the feature space (defined by wavelet approximation coefficients) iteratively by minimizing an objective function that represents weighted distance of a data point to a cluster centre. The weight assignment is done according to the membership grade of the data point according to some distance measure from the cluster centre. Both the membership grades and the cluster centers are iteratively updated to minimize the objective function [17, 18].

The fuzzy inference system (FIS) rules are determined from the final cluster centers and membership grades of the data points. It is done in the supervised manner by using the training dataset. The complete data set after wavelet decomposition consists of  $30 \times 7 = 210$  data vectors (30 samples each for 7 vapor classes) of 11 dimensions (wavelet coefficients). Half of the data were used for training and the rest half for testing. In the training phase: the input data for *genfis3* consists of  $105 \times 11$  matrix (105 samples of 7-class vapors  $\times$  11 DWT approximation coefficients), and the output data was defined as  $105 \times 8$  matrix where the 8 columns contained the vapor concentration values for samples in the first column, and the next 7 columns contained class assignment (1 for true, 0 for false). The output of *genfis3* is an FIS model. The test data is evaluated by using Matlab function '*evalfis*' which takes the FIS model created by *genfis3* as its input. The output from *evalfis* is  $105 \times 8$  matrix whose first column is the predicted concentration, each of next columns is the predicted class membership grade. The class membership grades were then converted to crisp class label assignments according to highest grade value.



## 4 Simulation

The pseudo-code for the Matlab program is as follows.

```

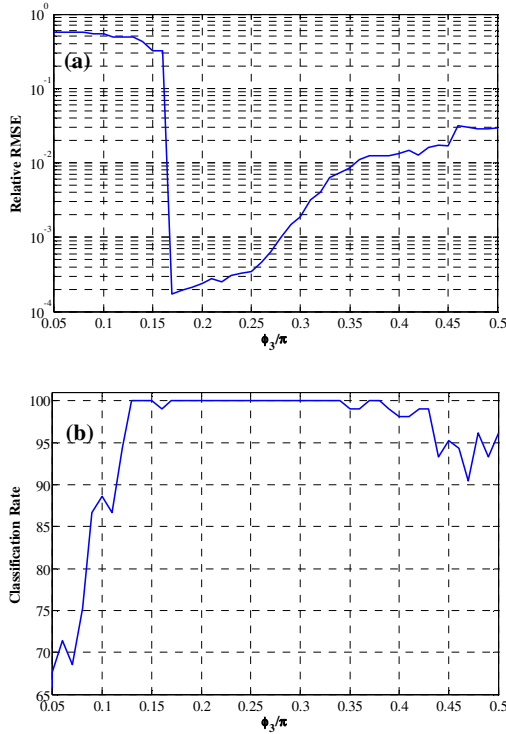
For  $\phi_3/\pi = 0.05$  to  $0.5$  step  $0.01$ 
Generate transient training and test data;
Transform transient data by DWT decomposition;
Define input data matrix (Xin) and output data matrix (Xout) for training;
Run fismat=genfis3(Xin, Xout, type, cluster n, options);
Run evalfis (Xtest, fismat) and obtain output Yout;
Convert fuzzy class membership grades in Yout to crisp class labels;
Calculate root mean squared fractional error for concentration estimation;
End.

```

The other parameters while using *genfis3* function were fixed as follows: no. of clusters `cluster_n=7`; options: (1) maximum number of iterations 300, and (2) minimum amount of improvement  $10^{-7}$ . The system performance metrics are taken to be the root mean square error in concentration estimates and percentage classification error. The FIS system performance was evaluated for variation in polymer film thickness over the range  $h = 22 - 1092$  nm equivalent to  $\phi_3/\pi = 0.01 - 0.5$  at interval of  $\Delta(\phi_3/\pi) = 0.01$ . This range of thickness variation extends from the acoustically thin linear to acoustically thick nonlinear film region of sensor response, approaching finally the film resonance condition for  $\phi_3/\pi = 0.5$  [9, 11]. This was done with an objective to examine the effect of polymer viscoelastic nonlinearity on the quantitative recognition of vapors, and determine the optimum film thickness.

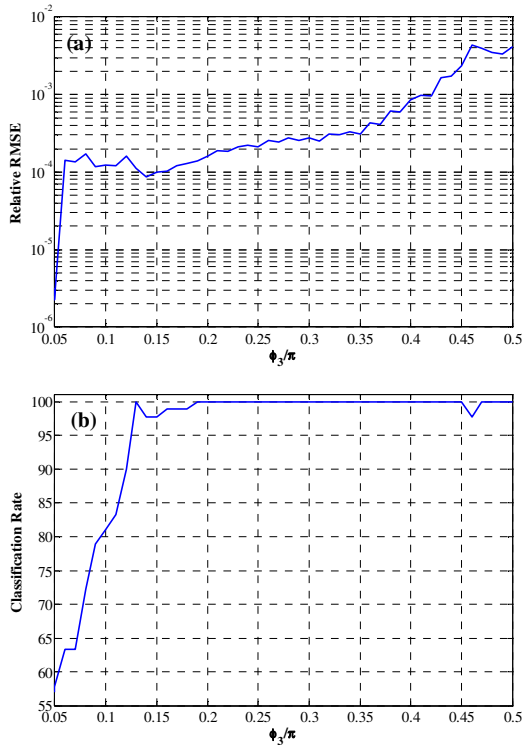
## 5 Results and Discussion

Fig 2 shows the quantification and classification results for the test samples (105 samples, 7 classes, 10-300 ppm concentration). The results are plotted with respect to polymer film thickness expressed in terms of  $\phi_3/\pi$ . In Fig 2(a) are shown the root mean squared fractional error in estimates of vapor concentration and in Fig 2(b) are displayed the correct classification rate in percentage. The mean squared error is determined by taking into accounts all 7 vapors for the complete range of concentrations. Note that in some intermediate region of film thickness ( $0.18 < \phi_3/\pi < 0.36$ ) the average RMS fractional error in concentration estimation is  $< 0.01$  and correct classification rate is 100%, and is the best region of performance. For thinner film condition ( $\phi_3/\pi < 0.18$ ) both the concentration and classification errors increase. In this region, the RMS fractional error in concentration increases to 0.6 and the classification rate reduces to 70%. On the other side, as the film resonance condition is approached at  $\phi_3/\pi = 0.5$  deteriorations in concentration estimate and classification rate are relatively much smaller. The former rises to 0.03 and the latter reduces to 90%. From this trend we conclude that an intermediate region of film thickness provides best results.



**Fig. 2.** (a) Root mean square of fractional error in concentration estimation of 7 volatile organics based on SAW sensor transients for concentrations over 10-300 ppm. The averaging is done over total samples of all the vapors. (b) Correct classification rate in percentage.

The deterioration of results at both ends (low thickness and resonant thickness) occurs for different reasons. In the low thickness region the normalized transient response curves for 6 vapors merged into nearly a single curve, except for dichlorobenzene which exhibited a different shape. This is expected because the diffusion coefficient for dichlorobenzene in polyisobutylene is  $5.49 \text{ cm}^2\text{s}^{-1}$ , which is very low compared to diffusion coefficient values for the other 6 vapors lying in the range  $35 \text{ to } 260 \text{ cm}^2\text{s}^{-1}$ . In the low thickness condition, all the fast diffusing species reach their steady state level quickly because the diffusion times  $\tau = h^2 / D$  for large diffusion coefficients  $D$  are small. As a result, distinction in shapes of transient curves for fast diffusing vapors is obliterated. Only the slow diffusing dichlorobenzene stands out. Another factor is that the partition coefficient for dichlorobenzene is highest at 22500 compared to the values over [180, 4680] for the others. Therefore, for any level of vapor concentration the signal dynamic range is quite large. In order to see the effect of large  $K$  and small  $D$  for dichlorobenzene the entire analysis was repeated after eliminating the dichlorobenzene data from the transient dataset. The results are shown in Fig. 3.



**Fig. 3.** Results after eliminating dichlorobenzene from the analysis, (a) RMS fractional error in concentration for remaining 6 volatile organics, and (b) classification rate in percentage

It can be seen that the RMS fractional error for concentration estimation has reduced to lower than 0.004 for all thicknesses, Fig 3(a). The classification result also improved to nearly 100% for all thicknesses beyond  $\phi_3/\pi = 0.19$ , Fig 3(b). It is noteworthy that deterioration near film resonance also decreased. The reason for this may be due to the negative effect of viscoelastic nonlinearity being more pronounced for high loading of dichlorobenzene (because of high  $K$ ). The effect of polymer viscoelasticity on vapor classification has been studied in detail by the authors, and reported in [7, 8, 13].

## 6 Conclusion

The results presented here demonstrate (i) the potentiality of SAW sensor transients in capturing both the quantitative and qualitative information about analyte vapors, and (ii) prove the efficacy of wavelet decomposition based feature extraction and  $c$ -means clustering based fuzzy inference system for simultaneous determination of vapor identity and concentration. It is shown that to make best of this methodology the polymer thickness must be optimized. The optimization should be to bring variability in the transient shapes for different vapors, and to keep away from the nonlinearity

near film resonance. For polyisobutylene coated SAW sensors analyzed in this paper, the optimum film thickness region corresponds to  $0.18 < \phi_3 / \pi < 0.36$ . In this region, the error in predicted vapor concentration is than 1% and the success in predicted class labels is 100%.

**Acknowledgments.** This work is supported by the Government of India, Defence Research & Development Organization Grant No. ERIP-ER-0703643-01-1025. The author PS is thankful to the CSIR/UGC, Government of India, for providing the SRF support for carrying out this work.

## References

1. Persaud, K., Dodd, G.: Analysis of Discrimination Mechanisms in the Mammalian Olfactory System Using a Model Nose. *Nature* 299, 352–355 (1982)
2. Rogers, E.K.: Handbook of Biosensors and Electronic Noses: Medicine, Food and Environment. CRC Press (1997)
3. Gardner, J.W., Bartlett, P.N.: Electronic Noses: Principles and Applications. Oxford University Press, New York (1999)
4. James, D., Simon, M.S., Ali, Z., William, T.H.: Chemical Sensors for Electronic Nose Systems. *Microchim. Acta* 149, 1–17 (2005)
5. Rock, F., Barsan, N., Weimar, U.: Electronic Nose: Current Status and Future Trends. *Chem. Rev.* 108, 705–725 (2008)
6. Alphas, D.W., Baietto, M.: Applications and Advances in Electronic-Nose Technologies. *Sensors* 9, 5099–5148 (2009)
7. Singh, P., Yadava, R.D.S.: Effect of Film Thickness and Viscoelasticity on Separability of Vapour Classes by Wavelet and Principal Component Analyses of Polymer-Coated Surface Acoustic Wave Sensor Transients. *Meas. Sci. Technol.* 22, 025202, 15 (2011)
8. Singh, P., Yadava, R.D.S.: Feature Extraction by Wavelet Decomposition of Surface Acoustic Wave Sensor Array Transients. *Def. Sci. J.* 60, 377–386 (2010)
9. Martin, S.J., Frye, G.C., Senturia, S.D.: Dynamics and Response of Polymer-Coated Surface Acoustic Wave Devices: Effect of Viscoelastic Properties and Film Resonance. *Anal. Chem.* 66, 2201–2219 (1994)
10. Yadava, R.D.S., Chaudhary, R.: Solvation, Transduction and Independent Component Analysis for Pattern Recognition in SAW Electronic Nose. *Sens. Actuat. B* 113, 1–21 (2006)
11. Yadava, R.D.S., Kshetrimayum, R., Khaneja, M.: Multifrequency Characterization of Viscoelastic Polymers and Vapor Sensing Based on SAW Oscillators. *Ultrasonics* 49, 638–645 (2009)
12. Crank, J.: *The Mathematics of Diffusion*. Clarendon, Oxford, sec. 4.3 eq. 4.18 (1986)
13. Singh, P., Yadava, R.D.S.: Using Parametric Nonlinearity in SAW Sensor Transients and Information Fusion for Improving Electronic Nose Intelligence. *Int. J. Computational Intelligence Research* 6, 919–927 (2010) (Special Conf. Issue ICCI 2010)
14. Singh, P., Yadava, R.D.S.: A Fusion Approach to Feature Extraction by Wavelet Decomposition and Principal Component Analysis in Transient Signal Processing of SAW Odor Sensor Array. *Sensors & Transducers J.* 126, 64–73 (2011)

15. Burrus, C.S., Gopinath, R.A., Guo, H.: Introduction to Wavelets and Wavelet Transforms: A Primer. Prentice-Hall, Englewood Cliffs (1998)
16. Mallat, S.: A Theory for Multiresolution Signal Decomposition: The Wavelet Representation. IEEE Pattern Anal. and Machine Intell. 11, 674–693 (1989)
17. Bezdek, J.C., Ehrlich, R., Full, W.: FCM: The Fuzzy  $c$ -Means Clustering Algorithm. Computers & Geosciences 10, 191–203 (1984)
18. Nascimento, S., Mirkin, B., Pires, F.: A Fuzzy Clustering Model of Data and Fuzzy  $c$ -Means. In: The Ninth IEEE International Conference on Fuzzy Systems, vol. 1, pp. 302–307 (2000)

# Author Index

- Abdelaziz, A.Y. I-679, II-257  
Afshar, Nejat A. II-201  
Agrawal, Sanjay I-159  
Ahmed, Faez II-71  
Al-Betar, Mohammed Azmi II-27, II-79  
Alia, Osama Moh'd II-79  
Anjana, A.V.R. II-267  
Arulmozhiyal, R. II-310  
Awadallah, Mohammed A. II-27  
  
Babawuro, Usman I-389  
Bajer, Drazen I-290  
Bakshi, Tuli I-381  
Balaga, Harish I-358  
Banerjee, Anuradha I-520  
Banerjee, Joydeep I-1  
Banerjee, Tribeni Prasad II-287  
Bapi, Raju S. II-166  
Basia, Pooja I-618  
Baskar, Subramanian I-77, I-282  
Baskaran, Kaliyaperumal II-310  
Benala, Tirimula Rao I-35, I-233  
Bharadwaj, Kamal K. I-433  
Bhargava, Mayank I-417  
Bhaskar, M. Arun I-135  
Bhattacharjee, Preetha I-601  
Bhattacharya, Bidishna I-68  
Bolaji, Asaju La'aro II-27  
Bose, Sandip II-105  
  
Chaitanya, Vedurupaka I-670  
Chakraborti, Tathagata II-89  
Chakraborty, Aruna I-460, I-610  
Chakraborty, Niladri I-68, I-151  
Chakraborty, Sumantra I-505  
Chaturvedi, D.K. I-494  
Chebrolu, Srilatha I-307  
Chen, Li-Yueh I-248  
Chittineni, Suresh II-211  
Choi, Byung-Jae I-469  
Chowdhury, Aritra II-105  
Chowdhury, Arkabandhu I-191  
Cui, Zhi-Hua II-132  
  
Dammavalam, Srinivasa Rao I-485  
Das, Apurba I-559  
Das, Asit Kumar I-372  
Das, Aveek Kumar I-94, I-110, I-119  
Das, Swagantam II-182  
Das, Swagatam I-1, I-19, I-51, I-94,  
I-102, I-670, I-688, II-105, II-223,  
II-239, II-287  
Dasgupta, Preetam I-19, I-27  
Dash, Subranshu Sekhar I-85, I-135,  
I-167  
De, Moumita II-55  
Deb, Kalyanmoy I-299, II-71  
Deepa, C. II-310  
Deepa, S.N. I-366  
Dehrouyeh, Mohammad Hadi I-407  
Dehuri, Satchidananda I-35, I-233  
Dehuri, S.N. II-9  
Devaraj, D. I-167  
Devi, B. Aruna I-366  
Devi, Swapna I-127  
Dhanalakshmi, Sundararajan I-282  
Dheeba, J. I-349  
Dhivya, Manian II-140  
Dinesh, G. II-211  
Dora, Lingaraj I-159  
Durga Bhavani, S. II-166  
Dutta, Paramartha I-520  
  
Ekbal, Asif I-425, II-231  
El-Khodary, S.M. I-679, II-257  
Fazendeiro, Paulo II-63  
  
Garg, Ritu I-183  
Gaurav, Raj II-46  
Geetha, T.V. I-530  
Geethanjali, M. II-267  
Ghosh, Ankur I-102  
Ghosh, Kuntal I-559  
Ghosh, Pradipta I-1, I-199  
Ghosh, Saurav II-182, II-223, II-239  
Ghosh, Subhankar I-520  
Girdhar, Isha I-618  
Gireeshkumar, T. II-294  
Golub, Marin I-662  
Gupta, H.M. I-217  
Gupta, Rohan I-417

- Halder, Anisha I-460, I-610  
 Halder, Udit I-19, I-27  
 Hanmandlu, Madasu I-217  
 Hannah, M. Esther I-530  
 Hasanuzzaman, Mohammad II-231  
 Hassanzadeh, Tahereh II-174  
 Holt, David I-452  
 Hong, Wei-Chiang I-248  
 Hui, Wu-Yin I-469
- Imran, Mohammad I-539  
 Inoussa, Garba I-389  
 Islam, Sk. Minhazul II-182, II-223,  
 II-239  
 Ivkovic, Nikola I-662
- Jadhav, Devidas G. I-127  
 Jain, Amit I-626  
 Jain, Anuj I-399  
 Jain, Himanshu I-299  
 Jain, Nitish II-46  
 Jana, Nanda Dulal I-209  
 Janarthanan, Ramadoss I-460, I-505,  
 I-601, I-610, II-89  
 Jasper, J. I-577  
 Javadikia, Payam I-407  
 Jayaprada, S. II-157  
 Jindal, Abhilash II-71
- Kabat, Manas Ranjan II-38  
 Kannan, Subramanian I-77, I-282  
 Kannan, V. II-267  
 Kannapiran, B. I-341  
 Kant, Vibhor I-433  
 Khader, Ahamad Tajudin II-27, II-79  
 Khosa, Rakesh I-714  
 Konar, Amit I-460, I-505, I-601, I-610,  
 II-89  
 Krishnanand, K.R. I-85, I-697  
 Krishna Prasad, M.H.M. I-485  
 Kshirsagar, Vivek II-113  
 Kumar, Amioy I-217, I-417  
 Kumar, Arun I-274  
 Kumar, Dirisala J. Nagendra I-315  
 Kumar, Gaurav II-46  
 Kumar, M. Jagadeesh I-135  
 Kumar, Piyush I-399  
 Kumar, Pradeep I-143  
 Kumar, Pravesh I-11  
 Kuppa, Mrithyumjaya Rao I-539
- Laha, Koushik I-102  
 Leno, I. Jerin I-323  
 Lewicki, Arkadiusz I-637, I-645  
 Lorestani, Ali Nejat I-407  
 Lotfi, Shahriar I-240, II-1
- Maddala, Seetha I-485  
 Mahadevan, Krishnan I-282  
 Maheswaran, Rathinasamy I-714  
 Mahmood, Ali Mirza I-539  
 Mahmoudi, Fariborz II-174  
 Maity, Dipankar I-19, I-27  
 Maji, Pradipta I-477  
 Majumdar, Ratul I-94, I-102, I-110  
 Majumder, Amit II-231  
 Majumder, Sibsankar I-151  
 Malekovic, Mirko I-662  
 Malik, L.G. I-265  
 Malik, O.P. I-494  
 Mallayya, Deivamani I-332  
 Mandal, Ankush I-119, I-199  
 Mandal, Kamal K. I-68, I-151  
 Mandal, Rajshree I-460  
 Mandava, Rajeswari I-79  
 Manikandan, R. II-191  
 Martinovic, Goran I-290  
 Mathur, Shashi I-731  
 Maulik, Ujjwal II-55  
 Meena, Yogesh Kumar II-302  
 Mehrotra, Kishan G. I-723  
 Mini, S. I-654  
 Mishra, Krishna K. I-274  
 Mitra, Anirban II-9  
 Mohan, Chilukuri K. I-723  
 Mohan, Yogeswaran II-17  
 Mohanta, Dushmantha K. I-706  
 Mohapatra, Ankita I-697  
 Moirangthem, Joymala I-85  
 Mondal, Arnab Kumar I-688  
 Mukherjee, Prithwijit I-119  
 Mukherjee, Saswati I-530  
 Mukhopadhyay, Anirban II-55  
 Murthy, J.V.R. I-315  
 Murthy, Pallavi I-176  
 Mutyalarao, M. II-122
- Naderloo, Leila I-407  
 Naegi, Sujata I-550  
 Nagori, Meghana II-113  
 Naik, Anima II-148

- Narahari Sastry, G. II-166  
 Nasir, Md. I-688  
 Nath, Hiran V. II-294  
 Nayak, Niranjan I-441  
  
 Osama, Reham A. I-679, II-257  
  
 Padmanabhan, B. I-577  
 Padmini, S. I-176  
 Pagadala, Aditya I-35  
 Pancerz, Krzysztof I-637, I-645  
 Panda, Ashok Kumar II-9  
 Panda, Rutuparna I-159  
 Panda, Sidhartha I-59  
 Pandit, Manjaree I-585  
 Pandit, Nicole I-585  
 Panigrahi, Bijaya Ketan I-85, I-110,  
 I-191, I-248, I-417, I-679, I-697, I-731,  
 II-257  
 Pant, Millie I-11, I-593  
 Patel, Manoj Kumar II-38  
 Patel, Rahila I-265  
 Patra, Gyana Ranjan I-51  
 Patra, Moumita II-248  
 Patra, M.R. II-9  
 Pattnaik, Shyam S. I-127  
 Paul, Sushmita I-477  
 Peddi, Santhosh I-225  
 Perkins, Louise I-452  
 Perumal, Krish II-46  
 Phadikar, Santanu I-372  
 Ponnambalam, S.G. I-43, I-323, II-17  
 Pothiraj, Sivakumar I-569  
 Potluri, Anupama II-97  
 Potti, Subbaraj I-569  
 Pradeep, A.N.S. II-211  
 Prasad, Shitala I-399  
 Prasad Reddy, P.V.G.D. II-211  
 Prata, Paula II-63  
 Pullela, S.V.V.S.R. Kumar I-315  
  
 Rabbani, Hekmat I-407  
 Raghavi, Ch. Sudha I-233  
 Raghuvanshi, M.M. I-265  
 Raha, Souvik I-102  
 Rahmani, Adel T. II-201  
 Raj, M. Victor I-323  
 Rajan, C. Christoher Asir I-176  
 Rajasekhar, Anguluri I-670  
 Rajesh, Vemulakonda I-539  
  
 Rakshit, Pratyusha I-601, I-610  
 Ramachandran, Baskaran I-332  
 Ramesh, Subramanian I-77  
 Ramezani, Fatemeh I-240  
 Rani, Manju II-302  
 Rao, Nalluri Madhusudana I-706  
 Ravindra Reddy, B. II-166  
 Reddy, S. Surender I-110  
 Roselyn, J. Preetha I-167  
 Rout, Pravat Kumar I-441, I-697  
 Routray, Sangram Kesari I-441  
 Roy, Anirban I-559  
 Roy, Diptendu Sinha I-706  
 Roy, Subhrajit II-182, II-223, II-239  
  
 Sabarinath, A. II-122  
 Sabat, Samrat L. I-654  
 Sadhu, Arup Kumar I-601  
 Saha, Nilanjan II-191  
 Saha, Sanchita I-425  
 Saha, Sriparna I-425, II-231  
 Salma, Umme II-278  
 Sanap, Shilpa A. II-113  
 Sanjeevi, Sriram G. I-307  
 Sankar, S. Saravana I-323  
 Sanyal, Subir kumar I-381  
 Sarkar, Bijan I-381  
 Sarkar, Soham I-51  
 Satapathy, Suresh Chandra I-233,  
 I-315, II-148, II-211  
 Satuluri, Naganjaneyulu I-539  
 Selvi, S. Tamil I-349  
 Sengupta, Abhronil II-89  
 Sengupta, Soumyadip I-688  
 Sequeira, Pedro II-63  
 Shamizi, Sevin II-1  
 Shankar, Deepa D. II-294  
 Sharma, Bhuvnesh I-618  
 Sharma, Tarun Kumar I-593  
 Sharma, Vaibhav I-217  
 Shrivastava, Nitin Anand I-731  
 Si, Tapas I-209  
 Sil, Jaya I-209, I-372  
 Singh, Alok I-225, II-97  
 Singh, Alpna I-550  
 Singh, Asheesh K. I-143  
 Singh, Awadhesh Kumar I-183  
 Singh, Kumar Anurag I-626  
 Singh, Pramod Kumar I-626  
 Singh, Prashant II-319



- Singh, Raj Mohan I-742  
 Singh, V.P. I-11, I-593  
 Sinha, Amrita I-358  
 Sinharay, Arindam I-381  
 Siriseti, G.S. Surya Vamsi I-35  
 Sivakumar, R.S. I-577  
 Sobha Rani, T. II-166  
 Soryani, Mohsen II-201  
 Spansel, Steven I-452  
 Srinivasa Rao, V. II-157  
 Srivastava, Praveen Ranjan I-618, II-46  
 Subbaraj, P. I-341  
 Subramani, C. I-135  
 Sudheer, Ch. I-731  
 Suganthan, Ponnuthurai Nagaratnam  
 II-182, II-223, II-239  
 Sundarambal, Murugesan II-140  
 Tadeusiewicz, Ryszard I-637, I-645  
 Taneja, Monika I-618  
 Tapaswi, Shashikala I-585  
 Tiwari, Aruna I-550  
 Tripathi, Anshul I-585  
 Tripathi, Subhransu Sekhar I-59  
 Tripathy, Chita Ranjan II-38  
 Tudu, Bhimsen I-68, I-151  
 Udgata, Siba K. I-654, II-248  
 Umrao, Rahul I-494  
 Ungati, Jagan Mohan II-46  
 Vadla, Sangeetha I-618  
 Vaisakh, K. II-278  
 Vasavi, S. II-157  
 Verma, Gaurav I-274  
 Verma, Prabha I-257  
 Victoire, T. Aruldoss Albert I-577  
 Vincent, J. Oswald II-140  
 Vincent, Lui Wen Han I-43  
 Vishwakarma, D.N. I-358  
 Vivek, S. I-135  
 Vojodi, Hakimeh II-174  
 Xavier James Raj, M. II-122  
 Yadava, R.D.S. I-257, II-319  
 Yang, Chun-Xia II-132  
 Yenduri, Sumanth I-452  
 Zafar, Hamim I-1, I-191, I-199  
 Zhao, Shizheng II-182, II-223, II-239