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

LNCS 7076

Swarm, Evolutionary, and Memetic Computing

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

1 Part I

Springer

Lecture Notes in Computer Science 7076

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 I

Springer

Volume Editors

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

Ponnuthurai Nagaratnam 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 978-3-642-27171-7 e-ISBN 978-3-642-27172-4
DOI 10.1007/978-3-642-27172-4
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 "Evaluating 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
Zhihua 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, Mexcico
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. Omkar
Minlong Lin
Shih-Hsin Chen
Sasitharan Balasubramaniam
Aniruddha Basak
Shih-Hsin Chen
Fatih Tasgetiren
Soumyadip Roy
S. Sivananaithapermal
Borko Boskovic
Pugalenthi 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 I

Design of Two-Channel Quadrature Mirror Filter Banks Using Differential Evolution with Global and Local Neighborhoods 1
Pradipta Ghosh, Hamim Zafar, Joydeep Banerjee, and Swagatam Das

Differential Evolution with Modified Mutation Strategy for Solving Global Optimization Problems 11
Pravesh Kumar, Millie Pant, and V.P. Singh

Self-adaptive Cluster-Based Differential Evolution with an External Archive for Dynamic Optimization Problems 19
Udit Halder, Dipankar Maity, Preetam Dasgupta, and Swagatam Das

An Informative Differential Evolution with Self Adaptive Re-clustering Technique 27
Dipankar Maity, Udit Halder, and Preetam Dasgupta

Differential Evolution for Optimizing the Hybrid Filter Combination in Image Edge Enhancement 35
Tirimula Rao Benala, Satchidananda Dehuri, G.S. Surya Vamsi Sirisetti, and Aditya Pagadala

Scheduling Flexible Assembly Lines Using Differential Evolution 43
Lui Wen Han Vincent and S.G. Ponnambalam

A Differential Evolution Based Approach for Multilevel Image Segmentation Using Minimum Cross Entropy Thresholding 51
Soham Sarkar, Gyana Ranjan Patra, and Swagatam Das

Tuning of Power System Stabilizer Employing Differential Evolution Optimization Algorithm 59
Subhransu Sekhar Tripathi and Sidhartha Panda

Logistic Map Adaptive Differential Evolution for Optimal Capacitor Placement and Sizing 68
Kamal K. Mandal, Bidishna Bhattacharya, Bhimsen Tudu, and Niladri Chakraborty

Application of an Improved Generalized Differential Evolution Algorithm to Multi-objective Optimization Problems 77
Subramanian Ramesh, Subramanian Kannan, and Subramanian Baskar

Enhanced Discrete Differential Evolution to Determine Optimal Coordination of Directional Overcurrent Relays in a Power System 85
Joymala Moirangthem, Subranshu Sekhar Dash, K.R. Krishnanand, and Bijaya Ketan Panigrahi

Dynamic Thinning of Antenna Array Using Differential Evolution Algorithm 94
Ratul Majumdar, Aveek Kumar Das, and Swagatam Das

A Quantized Invasive Weed Optimization Based Antenna Array Synthesis with Digital Phase Control 102
Ratul Majumdar, Ankur Ghosh, Souvik Raha, Koushik Laha, and Swagatam Das

Optimal Power Flow for Indian 75 Bus System Using Differential Evolution 110
Aveek Kumar Das, Ratul Majumdar, Bijaya Ketan Panigrahi, and S. Surender Reddy

A Modified Differential Evolution Algorithm Applied to Challenging Benchmark Problems of Dynamic Optimization 119
Ankush Mandal, Aveek Kumar Das, and Prithwijit Mukherjee

PSO Based Memetic Algorithm for Unimodal and Multimodal Function Optimization 127
Swapna Devi, Devidas G. Jadhav, and Shyam S. Pattnaik

Comparison of PSO Tuned Feedback Linearisation Controller (FBLC) and PI Controller for UPFC to Enhance Transient Stability 135
M. Jagadeesh Kumar, Subranshu Sekhar Dash, M. Arun Bhaskar, C. Subramani, and S. Vivek

A Nelder-Mead PSO Based Approach to Optimal Capacitor Placement in Radial Distribution System 143
Pradeep Kumar and Asheesh K. Singh

Comparative Performance Study of Genetic Algorithm and Particle Swarm Optimization Applied on Off-grid Renewable Hybrid Energy System 151
Bhimsen Tudu, Sibsankar Majumder, Kamal K. Mandal, and Niladri Chakraborty

An Efficient Algorithm for Multi-focus Image Fusion Using PSO-ICA ... 159
Sanjay Agrawal, Rutuparna Panda, and Lingaraj Dora

Economic Emission OPF Using Hybrid GA-Particle Swarm Optimization 167
J. Preetha Roselyn, D. Devaraj, and Subranshu Sekhar Dash

Application of Improved PSO Technique for Short Term Hydrothermal Generation Scheduling of Power System ... 176
S. Padmini, C. Christober Asir Rajan, and Pallavi Murthy

Multi-objective Workflow Grid Scheduling Based on Discrete Particle Swarm Optimization ... 183
Ritu Garg and Awadhesh Kumar Singh

Solution of Economic Load Dispatch Problem Using Lbest-Particle Swarm Optimization with Dynamically Varying Sub-swarms ... 191
Hamim Zafar, Arkabandhu Chowdhury, and Bijaya Ketan Panigrahi

Modified Local Neighborhood Based Niching Particle Swarm Optimization for Multimodal Function Optimization ... 199
Pradipta Ghosh, Hamim Zafar, and Ankush Mandal

Constrained Function Optimization Using PSO with Polynomial Mutation ... 209
Tapas Si, Nanda Dulal Jana, and Jaya Sil

Rank Based Hybrid Multimodal Fusion Using PSO ... 217
Amioy Kumar, Madasu Hanmandlu, Vaibhav Sharma, and H.M. Gupta

Grouping Genetic Algorithm for Data Clustering ... 225
Santhosh Peddi and Alok Singh

Genetic Algorithm for Optimizing Neural Network Based Software Cost Estimation ... 233
Tirimula Rao Benala, Satchidananda Dehuri, Suresh Chandra Satapathy, and Ch. Sudha Raghavi

IAMGA: Intimate-Based Assortative Mating Genetic Algorithm ... 240
Fatemeh Ramezani and Shahriar Lotfi

SVR with Chaotic Genetic Algorithm in Taiwanese 3G Phone Demand Forecasting ... 248
Li-Yueh Chen, Wei-Chiang Hong, and Bijaya Ketan Panigrahi

Genetic Algorithm Assisted Enhancement in Pattern Recognition Efficiency of Radial Basis Neural Network ... 257
Prabha Verma and R.D.S. Yadava

An Approach Based on Grid-Value for Selection of Parents in Multi-objective Genetic Algorithm ... 265
Rahila Patel, M.M. Raghuwanshi, and L.G. Malik

A Novel Non-dominated Sorting Algorithm ... 274
Gaurav Verma, Arun Kumar, and Krishna K. Mishra

Intelligent Genetic Algorithm for Generation Scheduling under Deregulated Environment 282
Sundararajan Dhanalakshmi, Subramanian Kannan, Subramanian Baskar, and Krishnan Mahadevan

Impact of Double Operators on the Performance of a Genetic Algorithm for Solving the Traveling Salesman Problem 290
Goran Martinovic and Drazen Bajer

Parent to Mean-Centric Self-Adaptation in SBX Operator for Real-Parameter Optimization 299
Himanshu Jain and Kalyanmoy Deb

Attribute Reduction in Decision-Theoretic Rough Set Models Using Genetic Algorithm 307
Srilatha Chebrolu and Sriram G. Sanjeevi

A Study of Decision Tree Induction for Data Stream Mining Using Boosting Genetic Programming Classifier 315
Dirisala J. Nagendra Kumar, J.V.R. Murthy, Suresh Chandra Satapathy, and S.V.V.S.R. Kumar Pullela

Bi-criteria Optimization in Integrated Layout Design of Cellular Manufacturing Systems Using a Genetic Algorithm 323
I. Jerin Leno, S. Saravana Sankar, M. Victor Raj, and S.G. Ponnambalam

Reconfigurable Composition of Web Services Using Belief Revision through Genetic Algorithm 332
Deivamani Mallayya and Baskaran Ramachandran

Neural Network Based Model for Fault Diagnosis of Pneumatic Valve with Dimensionality Reduction 341
P. Subbaraj and B. Kannapiran

A CAD System for Breast Cancer Diagnosis Using Modified Genetic Algorithm Optimized Artificial Neural Network 349
J. Dheeba and S. Tamil Selvi

Application of ANN Based Pattern Recognition Technique for the Protection of 3-Phase Power Transformer 358
Harish Balaga, D.N. Vishwakarma, and Amrita Sinha

Modified Radial Basis Function Network for Brain Tumor Classification 366
S.N. Deepa and B. Aruna Devi

Attribute Clustering and Dimensionality Reduction Based on In/Out Degree of Attributes in Dependency Graph 372
Asit Kumar Das, Jaya Sil, and Santanu Phadikar

MCDM Based Project Selection by F-AHP & VIKOR 381
Tuli Bakshi, Arindam Sinharay, Bijan Sarkar, and Subir kumar Sanyal

Nonlinear Time Series Modeling and Prediction Using Local Variable Weights RBF Network 389
Garba Inoussa and Usman Babawuro

Detection of Disease Using Block-Based Unsupervised Natural Plant Leaf Color Image Segmentation 399
Shitala Prasad, Piyush Kumar, and Anuj Jain

Measuring the Weight of Egg with Image Processing and ANFIS Model 407
Payam Javadikia, Mohammad Hadi Dehrouyeh, Leila Naderloo, Hekmat Rabbani, and Ali Nejat Lorestani

Palmprint Authentication Using Pattern Classification Techniques 417
Amioy Kumar, Mayank Bhargava, Rohan Gupta, and Bijaya Ketan Panigrahi

A Supervised Approach for Gene Mention Detection 425
Sriparna Saha, Asif Ekbal, and Sanchita Saha

Incorporating Fuzzy Trust in Collaborative Filtering Based Recommender Systems 433
Vibhor Kant and Kamal K. Bharadwaj

A Function Based Fuzzy Controller for VSC-HVDC System to Enhance Transient Stability of AC/DC Power System 441
Niranjan Nayak, Sangram Kesari Routray, and Pravat Kumar Rout

A Bayesian Network Riverine Model Study 452
Steven Spansel, Louise Perkins, Sumanth Yenduri, and David Holt

Application of General Type-2 Fuzzy Set in Emotion Recognition from Facial Expression 460
Anisha Halder, Rajshree Mandal, Aruna Chakraborty, Amit Konar, and Ramadoss Janarthanan

Design of a Control System for Hydraulic Cylinders of a Sluice Gate Using a Fuzzy Sliding Algorithm 469
Wu-Yin Hui and Byung-Jae Choi

Rough Sets for Selection of Functionally Diverse Genes from Microarray Data 477
Sushmita Paul and Pradipta Maji

Quality Evaluation Measures of Pixel - Level Image Fusion Using Fuzzy Logic 485
Srinivasa Rao Dammavalam, Seetha Maddala, and M.H.M. Krishna Prasad

Load Frequency Control: A Polar Fuzzy Approach 494
Rahul Umrao, D.K. Chaturvedi, and O.P. Malik

An Efficient Algorithm to Computing Max-Min Post-inverse Fuzzy Relation for Abductive Reasoning 505
Sumantra Chakraborty, Amit Konar, and Ramadoss Janarthanan

Fuzzy-Controlled Energy-Efficient Weight-Based Two Hop Clustering for Multicast Communication in Mobile Ad Hoc Networks 520
Anuradha Banerjee, Paramartha Dutta, and Subhankar Ghosh

Automatic Extractive Text Summarization Based on Fuzzy Logic: A Sentence Oriented Approach 530
M. Esther Hannah, T.V. Geetha, and Saswati Mukherjee

An Improved CART Decision Tree for Datasets with Irrelevant Feature 539
Ali Mirza Mahmood, Mohammad Imran, Naganjaneyulu Satuluri, Mrithyumjaya Rao Kuppa, and Vemulakonda Rajesh

Fuzzy Rough Set Approach Based Classifier 550
Alpna Singh, Aruna Tiwari, and Sujata Naegi

Proposing a CNN Based Architecture of Mid-level Vision for Feeding the WHERE and WHAT Pathways in the Brain 559
Apurba Das, Anirban Roy, and Kuntal Ghosh

Multithreaded Memetic Algorithm for VLSI Placement Problem 569
Subbaraj Potti and Sivakumar Pothiraj

Bacterial Foraging Approach to Economic Load Dispatch Problem with Non Convex Cost Function 577
B. Padmanabhan, R.S. Sivakumar, J. Jasper, and T. Aruldoss Albert Victoire

Static/Dynamic Environmental Economic Dispatch Employing Chaotic Micro Bacterial Foraging Algorithm 585
Nicole Pandit, Anshul Tripathi, Shashikala Tapaswi, and Manjaree Pandit

Artificial Bee Colony Algorithm with Self Adaptive Colony Size 593
Tarun Kumar Sharma, Millie Pant, and V.P. Singh

Multi-Robot Box-Pushing Using Non-dominated Sorting Bee Colony Optimization Algorithm 601
Pratyusha Rakshit, Arup Kumar Sadhu, Preetha Bhattacharjee, Amit Konar, and Ramadoss Janarthanan

Emotion Recognition from the Lip-Contour of a Subject Using Artificial Bee Colony Optimization Algorithm 610
Anisha Halder, Pratyusha Rakshit, Aruna Chakraborty, Amit Konar, and Ramadoss Janarthanan

Software Coverage : A Testing Approach through Ant Colony Optimization 618
Bhuvnesh Sharma, Isha Girdhar, Monika Taneja, Pooja Basia, Sangeetha Vadla, and Praveen Ranjan Srivastava

Short Term Load Forecasting Using Fuzzy Inference and Ant Colony Optimization 626
Amit Jain, Pramod Kumar Singh, and Kumar Anurag Singh

The Use of Strategies of Normalized Correlation in the Ant-Based Clustering Algorithm 637
Arkadiusz Lewicki, Krzysztof Pancerz, and Ryszard Tadeusiewicz

Ant Based Clustering of Time Series Discrete Data – A Rough Set Approach 645
Krzysztof Pancerz, Arkadiusz Lewicki, and Ryszard Tadeusiewicz

Sensor Deployment for Probabilistic Target k-Coverage Using Artificial Bee Colony Algorithm 654
S. Mini, Siba K. Udgata, and Samrat L. Sabat

Extended Trail Reinforcement Strategies for Ant Colony Optimization 662
Nikola Ivkovic, Mirko Malekovic, and Marin Golub

Fractional-Order $\mathrm{PI}^{\lambda}\mathrm{D}^{\mu}$ Controller Design Using a Modified Artificial Bee Colony Algorithm 670
Anguluri Rajasekhar, Vedurupaka Chaitanya, and Swagatam Das

Reconfiguration of Distribution Systems for Loss Reduction Using the Harmony Search Algorithm 679
A.Y. Abdelaziz, Reham A. Osama, S.M. El-Khodary, and Bijaya Ketan Panigrahi

An Improved Multi-objective Algorithm Based on Decomposition with Fuzzy Dominance for Deployment of Wireless Sensor Networks 688
Soumyadip Sengupta, Md. Nasir, Arnab Kumar Mondal, and Swagatam Das

Application of Multi-Objective Teaching-Learning-Based Algorithm to an Economic Load Dispatch Problem with Incommensurable Objectives 697
K.R. Krishnanand, Bijaya Ketan Panigrahi, P.K. Rout, and Ankita Mohapatra

Application of NSGA – II to Power System Topology Based Multiple Contingency Scrutiny for Risk Analysis 706
Nalluri Madhusudana Rao, Diptendu Sinha Roy, and Dusmanta K. Mohanta

Multi Resolution Genetic Programming Approach for Stream Flow Forecasting 714
Rathinasamy Maheswaran and Rakesh Khosa

Reference Set Metrics for Multi-Objective Algorithms 723
Chilukuri K. Mohan and Kishan G. Mehrotra

Groundwater Level Forecasting Using SVM-QPSO 731
Ch. Sudheer, Nitin Anand Shrivastava, Bijaya Ketan Panigrahi, and M Shashi Mathur

Genetic Algorithm Based Optimal Design of Hydraulic Structures with Uncertainty Characterization 742
Raj Mohan Singh

Author Index 751

Table of Contents – Part II

Register Allocation via Graph Coloring Using an Evolutionary Algorithm 1
Sevin Shamizi and Shahriar Lotfi

A Survey on Swarm and Evolutionary Algorithms for Web Mining Applications 9
Ashok Kumar Panda, S.N. Dehuri, M.R. Patra, and Anirban Mitra

Exploration Strategies for Learning in Multi-agent Foraging 17
Yogeswaran Mohan and S.G. Ponnambalam

Nurse Rostering Using Modified Harmony Search Algorithm 27
Mohammed A. Awadallah, Ahamad Tajudin Khader, Mohammed Azmi Al-Betar, and Asaju La'aro Bolaji

A Swarm Intelligence Based Algorithm for QoS Multicast Routing Problem 38
Manoj Kumar Patel, Manas Ranjan Kabat, and Chita Ranjan Tripathy

Test Data Generation: A Hybrid Approach Using Cuckoo and Tabu Search 46
Krish Perumal, Jagan Mohan Ungati, Gaurav Kumar, Nitish Jain, Raj Gaurav, and Praveen Ranjan Srivastava

Selection of GO-Based Semantic Similarity Measures through AMDE for Predicting Protein-Protein Interactions 55
Anirban Mukhopadhyay, Moumita De, and Ujjwal Maulik

Towards Cost-Effective Bio-inspired Optimization: A Prospective Study on the GPU Architecture 63
Paula Prata, Paulo Fazendeiro, and Pedro Sequeira

Cricket Team Selection Using Evolutionary Multi-objective Optimization 71
Faez Ahmed, Abhilash Jindal, and Kalyanmoy Deb

Data Clustering Using Harmony Search Algorithm 79
Osama Moh'd Alia, Mohammed Azmi Al-Betar, Rajeswari Mandava, and Ahamad Tajudin Khader

Application of Swarm Intelligence to a Two-Fold Optimization Scheme for Trajectory Planning of a Robot Arm 89
Tathagata Chakraborti, Abhronil Sengupta, Amit Konar, and Ramadoss Janarthanan

Two Hybrid Meta-heuristic Approaches for Minimum Dominating Set Problem 97
Anupama Potluri and Alok Singh

Automatic Clustering Based on Invasive Weed Optimization Algorithm 105
Aritra Chowdhury, Sandip Bose, and Swagatam Das

Classification of Anemia Using Data Mining Techniques 113
Shilpa A. Sanap, Meghana Nagori, and Vivek Kshirsagar

Taboo Evolutionary Programming Approach to Optimal Transfer from Earth to Mars 122
M. Mutyalarao, A. Sabarinath, and M. Xavier James Raj

Solving Redundancy Optimization Problem with a New Stochastic Algorithm 132
Chun-Xia Yang and Zhi-Hua Cui

Energy Efficient Cluster Formation in Wireless Sensor Networks Using Cuckoo Search 140
Manian Dhivya, Murugesan Sundarambal, and J. Oswald Vincent

Data Clustering Based on Teaching-Learning-Based Optimization 148
Suresh Chandra Satapathy and Anima Naik

Extracting Semantically Similar Frequent Patterns Using Ontologies 157
S. Vasavi, S. Jayaprada, and V. Srinivasa Rao

Correlating Binding Site Residues of the Protein and Ligand Features to Its Functionality 166
B. Ravindra Reddy, T. Sobha Rani, S. Durga Bhavani, Raju S. Bapi, and G. Narahari Sastry

Non-linear Grayscale Image Enhancement Based on Firefly Algorithm 174
Tahereh Hassanzadeh, Hakimeh Vojodi, and Fariborz Mahmoudi

Synthesis and Design of Thinned Planar Concentric Circular Antenna Array - A Multi-objective Approach 182
Sk. Minhazul Islam, Saurav Ghosh, Subhrajit Roy, Shizheng Zhao, Ponnuthurai Nagaratnam Suganthan, and Swagamtam Das

Soft Computing Based Optimum Parameter Design of PID Controller in Rotor Speed Control of Wind Turbines 191
R. Manikandan and Nilanjan Saha

Curve Fitting Using Coevolutionary Genetic Algorithms 201
Nejat A. Afshar, Mohsen Soryani, and Adel T. Rahmani

A Parallel Hybridization of Clonal Selection with Shuffled Frog Leaping Algorithm for Solving Global Optimization Problems (P-AISFLA) 211
Suresh Chittineni, A.N.S. Pradeep, G. Dinesh, Suresh Chandra Satapathy, and P.V.G.D. Prasad Reddy

Non-uniform Circular-Shaped Antenna Array Design and Synthesis - A Multi-Objective Approach 223
Saurav Ghosh, Subhrajit Roy, Sk. Minhazul Islam, Shizheng Zhao, Ponnuthurai Nagaratnam Suganthan, and Swagatam Das

Supervised Machine Learning Approach for Bio-molecular Event Extraction 231
Asif Ekbal, Amit Majumder, Mohammad Hasanuzzaman, and Sriparna Saha

Design of Two Channel Quadrature Mirror Filter Bank: A Multi-Objective Approach 239
Subhrajit Roy, Sk. Minhazul Islam, Saurav Ghosh, Shizheng Zhao, Ponnuthurai Nagaratnam Suganthan, and Swagatam Das

Soft Computing Approach for Location Management Problem in Wireless Mobile Environment 248
Moumita Patra and Siba K. Udgata

Distribution Systems Reconfiguration Using the Hyper-Cube Ant Colony Optimization Algorithm 257
A.Y. Abdelaziz, Reham A. Osama, S.M. El-Khodary, and Bijaya Ketan Panigrahi

Bacterial Foraging Optimization Algorithm Trained ANN Based Differential Protection Scheme for Power Transformers 267
M. Geethanjali, V. Kannan, and A.V.R. Anjana

Reduced Order Modeling of Linear MIMO Systems Using Soft Computing Techniques 278
Umme Salma and K. Vaisakh

Statistical and Fusion Based Hybrid Approach for Fault Signal Classification in Electromechanical System 287
Tribeni Prasad Banerjee and Swagatam Das

Steganalysis for Calibrated and Lower Embedded Uncalibrated Images 294
Deepa D. Shankar, T. Gireeshkumar, and Hiran V. Nath

An Efficient Feature Extraction Method for Handwritten Character Recognition 302
Manju Rani and Yogesh Kumar Meena

Optimized Neuro PI Based Speed Control of Sensorless Induction Motor 310
R. Arulmozhiyal, C. Deepa, and Kaliyaperumal Baskaran

Wavelet Based Fuzzy Inference System for Simultaneous Identification and Quantitation of Volatile Organic Compounds Using SAW Sensor Transients 319
Prashant Singh and R.D.S. Yadava

Author Index 329

Design of Two-Channel Quadrature Mirror Filter Banks Using Differential Evolution with Global and Local Neighborhoods

Pradipta Ghosh, Hamim Zafar, Joydeep Banerjee, and Swagatam Das

Electronics & Tele-Comm. Engineering
Jadavpur University
Kolkata, India
{iampradiptaghosh,hmm.zafar,iamjoydeepbanerjee}@gmail.com,
swagatamdas19@yahoo.co.in

Abstract. This paper introduces a novel method named DEGL (Differential Evolution with global and local neighborhoods) regarding the design of two channel quadrature mirror filter with linear phase characteristics. To match the ideal system response characteristics, this improved variant of Differential Evolution technique is employed to optimize the values of the filter bank coefficients. The filter response is optimized in both pass band and stop band. The overall filter bank response consists of objective functions termed as reconstruction error, mean square error in pass band and mean square error in stop band. Effective designing can be performed if the objective function is properly minimized. The proposed algorithm can perform much better than the other existing design methods. Three different design examples are presented here for the illustrations of the benefits provided by the proposed algorithm.

Keywords: Filter banks, Quadrature Mirror Filter, Sub-band coding, perfect reconstruction, DEGL.

1 Introduction

Efficient design of filter banks has become a promising area of research work. An improved design of filter can have significant effect on different aspects of signal processing and many fields such as speech coding, scrambling, image processing, and transmission of several signals through same channel [1]. Among various filter banks, the two channel QMF bank was first used in Sub-band coding, in which the signal is divided into several frequency bands and digital encoders of each band of signal can be used for analysis. QMF also finds application in representation of signals in digital form for transmission and storage purpose in speech processing, image processing and its compression, communication systems, power system networks, antenna systems [2], analog to digital (A/D) converter [3], and design of wavelet base [4].

Various optimizations based or non optimization based design techniques for QMF have been found in literature. Recently several efforts have been made for designing the optimized QMF banks based on linear and non-linear phase objective function using various evolutionary algorithms. Various methods such as least square technique [5-7],

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 1–10, 2011.
© Springer-Verlag Berlin Heidelberg 2011

weighted least square (WLS) technique [8-9] have been applied to solve the problem. But due to high degree of nonlinearity and complex optimization technique, these methods were not suitable for the filter with larger taps. A method based on eigenvector computation in each iteration is proposed to obtain the optimum quantized filter weights [10]. An optimization method based on Genetic algorithm and signed-power-of-two [11] is successfully applied in designing the lattice QMF. In frequency domain methods reconstruction error is not equiripple [7-9]. Chen and Lee have proposed an iterative technique [8] that results in equiripple reconstruction error, and the generalization of this method was carried out in [9] to obtain equiripple behaviors in stop band. Unfortunately, these techniques are complicated, and are only applicable to the two-band QMF banks that have low orders. To solve the previous problems, a two-step approach for the design of two-channel filter banks [12,13] was developed. But this approach results in nonlinear phase, and is not suitable for the wideband audio signal. A more robust and powerful tool PSO has also been applied for the design of the optimum QMF bank with reduced aliasing distortion, phase aliasing and reconstruction errors [14,15]. The problem with PSO is the premature convergence due to the presence of local optima.

Amongst all Evolutionary Algorithms (EAs) described in various articles, Differential Evolution (DE) has emerged as one of the most powerful tools for solving the real world optimization problems. It has not been applied on the design of the perfectly reconstructed QMF banks till now. In this context we present here a new powerful variant of DE called DEGL [20] for the efficient design of two channels QMF bank. Later in this paper, we will discuss the effectiveness of this algorithm and compare this with other existing method towards designing of Digital QMF filter. For proving our point we have presented three different design problems.

The rest of the paper is arranged in the following way: Section 2 contains the Design Problem, Section 3 gives a brief overview of classical DE, Section 4 introduces a new improved variant of DE termed as DEGL, Section 5 deals with the design results and comparison of these results with other algorithms and Section 6 concludes this paper.

2 Formulation of Design Problems

For a typical two-channel QMF bank as shown in Fig. 1 the reconstructed output signal is defined as

$$\begin{aligned} Y(z) &= 1/2[H_0(z)G_0(z)+H_1(z)G_1(z)]X(z)+1/2[H_0(-z)G_0(z)+H_1(-z)G_1(z)]X(-z) \\ &= T(z)X(z)+A(z)X(-z) \end{aligned} \tag{1}$$

where $Y(z)$ is the reconstructed signal and $X(z)$ is the original signal.

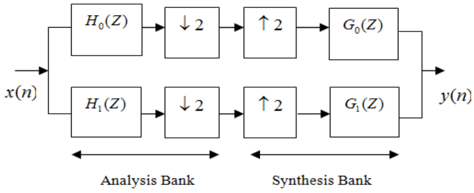


Fig. 1. Two channel QMF BANK

In eqn.1 the first term $\mathrm{T(z)}$ is the desired translation from input to the output, called distortion transfer function, while second term, $\mathrm{A(z)}$ is aliasing distortion because of change in sampling rate. By setting

$$H_1(z) = H_0(-z), G_0(z) = H_1(-z) \text{ and } G_1(z) = -H_0(-z) \tag{2}$$

the aliasing error in (1) can be completely eliminated. Then eqn. (1) becomes

$$Y(z) = 1/2[H_0(z)H_0(z) + H_0(-z)H_0(-z)]X(z) \tag{3}$$

It implies that the overall design problem of filter bank reduces to determination of the filter taps coefficients of a low pass filter $\mathrm{H_0(z)}$, called a prototype filter. Let $\mathrm{H_0(z)}$ be a linear phase finite impulse response (FIR) filter with even length (N):

$$H_0(e^{j\omega}) = \left|H_0(e^{j\omega})\right| e^{-j\omega(N-1)/2} \tag{4}$$

If all above mentioned conditions are put together, the overall transfer function of QMF bank is reduced to Eqn.(5) $T(e^{j\omega}) = \frac{e^{-j\omega(N-1)/2}}{2}\left\{\left|H_0(e^{j\omega})\right|^2 + \left|H_0(e^{j(\omega-\pi})\right|^2\right\}$ (5)

$$T(e^{j\omega}) = \frac{1}{2} e^{-j\omega(N-1)/2} T'(e^{j\omega}) \tag{6}$$

Where, $T'(e^{j\omega}) = [\left|H_0(\omega)\right|^2 - (-1)^{(N-1)}\left|H_0(\pi-\omega)\right|^2]$ and N is the no of filters.

If $\left|H_0(e^{j\omega})\right|^2 + \left|H_0(e^{j(\omega-\pi})\right|^2 = 1$ it results in perfect reconstruction; the output signal is exact replica of the input signal. If it is evaluated at frequency ($\omega = 0.5\pi$), the perfect reconstruction condition reduces to Eqn. (7).

$$\left|H_0(e^{j0.5\pi})\right| = 0.707 \tag{7}$$

It shows that QMF bank has a linear phase delay due to the term $e^{-j\omega(N-1)/2}$.

Then the condition for the perfect reconstruction is to minimize the weighted sum of four terms as shown below:

$$K = \alpha_1 E_p + \alpha_2 E_s + \alpha_3 E_t + \alpha_4 .mor + \alpha_5 E_h \tag{8}$$

where $\alpha_1 - \alpha_5$ are the relative weights and E_p , E_s , E_t , mor (measure of ripple) , E_h are defined as follows: E_p is the mean square error in pass band (MSEP) which describes the energy of reconstruction error between 0 and ω_p ,

$$E_p = \frac{1}{\pi}\int_0^{\omega} \left|H_0(0) - H_0(\omega)\right|^2 d\omega \tag{9}$$

E_s is the mean square error in stop band (MSES) which denotes the stop band energy related to LPF between ω_s to π

$$E_s = \frac{1}{\pi}\int_{\omega_s}^{\pi} \left|H_0(\omega)\right|^2 d\omega \tag{10}$$

E_t is the square error of overall transfer function at quadrature frequency $\pi/2$

$$E_t = [H_0(\frac{\pi}{2}) - \frac{1}{\sqrt{2}} H_0(0)]^2 \tag{11}$$

Measure of ripple (mor)

$$mor = \max_{\omega}\left|10\log_{10}\left|T'(\omega)\right|\right| - \min_{\omega}\left|10\log_{10}\left|T'(\omega)\right|\right| \tag{12}$$

E_h is the deviation of $T'(e^{j\omega})$ from unity at $\omega = \pi/2$

$$E_h = \left|T'(\frac{\pi}{2}) - 1\right| \tag{13}$$

The above fitness is considered because we know that condition for perfect reconstruction filter is

$$\left|H_0(e^{j\omega})\right|^2 + \left|H_0(e^{j(\omega-\pi})\right|^2 = 1 \tag{14}$$

3 DE with Global and Local Neighborhoods (DEGL)

DE is a simple real-coded evolutionary algorithm [16]. It works through a simple cycle of stages, which are detailed in [16]. In this section we describe the variant of DE termed as DEGL. Suppose we have a DE population

$$P_G = [\vec{X}_{1,G}, \vec{X}_{2,G}, \vec{X}_{3,G}, ..., \vec{X}_{NP,G}] \tag{15}$$

Where each $\vec{X}_{i,G}$ $(i = 1,2,3,......,NP)$ a *D* dimensional parameter vector. Now, for every vector $\vec{X}_{i,G}$ we define a neighborhood of radius *k* (where *k* is a nonzero integer from 0 to (NP-1)/2, as the neighborhood size must be smaller than the population size, i.e. $2k + 1 \le NP$), consisting of vectors $\vec{X}_{i-k,G}, . . , \vec{X}_{i,G}, ..., \vec{X}_{i+k,G}$. We assume the vectors to be organized on a ring topology with respect to their indices, such that vectors $\vec{X}_{NP,G}$ and $\vec{X}_{2,G}$ are the two immediate neighbors of vector $\vec{X}_{1,G}$.For each member of the population, a local donor vector is created by employing the best (fittest) vector in the neighborhood of that member and any two other vectors chosen from the same neighborhood. The model may be expressed as

$$\vec{L}_{i,G} = \vec{X}_{i,G} + \alpha.(\vec{X}_{n\_best_i,G} - \vec{X}_{i,G}) + \beta.(\vec{X}_{p,G} - \vec{X}_{q,G}) \tag{16}$$

where the subscript *n_best*$_i$ indicates the best vector in the neighborhood of $\overrightarrow{X}_{i,G}$ and $p, q \in [i - k, i + k]$ with $p \neq q \neq i$. Similarly, the global donor vector is created as

$$\vec{g}_{i,G} = \vec{X}_{i,G} + \alpha.(\vec{X}_{n\_gbest,G} - \vec{X}_{i,G}) + \beta.(\vec{X}_{r1,G} - \vec{X}_{r2,G}) \tag{17}$$

where the subscript *g_best* indicates the best vector in the entire population at generation *G* and $r1, r2 \in [1, NP]$ with $r1 \neq r2 \neq i$. α and β are the scaling factors. Note that in (16) and (17), the first perturbation term on the right-hand side (the one multiplied by α) is an arithmetical recombination operation, while the second term (the one multiplied by β) is the differential mutation. Thus in both the global and local mutation models, we basically generate mutated recombinants, not pure mutants.

Now we combine the local and global donor vectors using a scalar weight $\omega \in (0,1)$ to form the actual donor vector of the proposed algorithm

$$\vec{V}_{i,G} = \omega.\vec{g}_{i,G} + (1-\omega)..\vec{L}_{i,G} \tag{18}$$

Clearly, if $\omega = 1$ and in addition $\alpha = \beta = F$, the donor vector generation scheme in (18) reduces to that of DE/target to-best/1. Hence the latter may be considered as a special case of this more general strategy involving both global and local neighborhood of each vector synergistically. From now on, we shall refer to this version as DEGL (DE with global and local neighborhoods). The rest of the algorithm is exactly similar to DE/rand/1/bin. DEGL uses a binomial crossover scheme.

3.1 Control Parameters in DEGL

DEGL introduces four new parameters. They are: α, β, ω and the neighborhood radius *k*. In order to lessen the number of parameters further, we take $\alpha = \beta = F$. The most important parameter in DEGL is perhaps the weight factor ω, which controls the balance between the exploration and exploitation capabilities. Small values of ω (close to 0) in (11) favor the local neighborhood component, thereby resulting in better exploration. There are three different schemes for the selection and adaptation of ω to gain intuition regarding DEGL performance. They are *Increasing weight Factor, Random Weight Factor, Self-Adaptive Weight Factor* respectively. But we have used only *Random Weight Factor* for this design problem. So we will describe only the incorporated method in the following paragraphs.

3.2 Random Weight Factor

In this scheme the weight factor of each vector is made to vary as a uniformly distributed random number in (0, 1) i.e. $\omega_{i,G} \approx rand(0,1)$. Such a choice may decrease the convergence speed (by introducing more diversity). But the minimum value is 0.15.

3.3 Advantage of Random Weight Factor

This scheme had empirically proved to be the best scheme among all three schemes defined in original DEGL article for this kind of design problem. The most important advantage in this scheme lies on the process of crossover. Due to varying weight factor the no of possible different vector increases. So the searching is much wider than using other two schemes.

4 Design Problems

4.1 Parameter Initializations

For the design purpose we set the searching upper bound = 0.5 and searching lower bound = -0.5; Function bound Constraint for DEGL is set to be 0. The initial population size is 100. The no of generations for DEGL is set equal to 500. Next we had to set the values of the constant terms i.e. $\alpha_1 - \alpha_5$ in Eqn. 8. For all the examples, the relative weights of fitness function are determined based on trial and error method using concepts of QMF filter. The values of the constants are as follows.

$\alpha_1 = .95,\ \alpha_2 = .07,\ \alpha_3 = .07,\ \alpha_4 = 10^{-4},\ \alpha_5 = 10^{-1}.$

4.2 Problem Examples

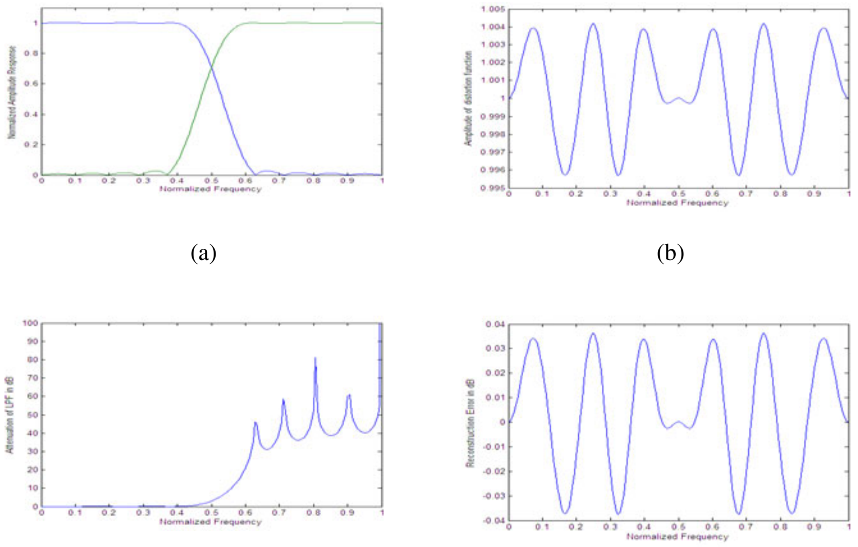
4.2.1 Two-Channel QMF Bank for N = 22, $\omega_p = 0.4\pi$, $\omega_s = 0.6\pi$, with 11 Filter Coefficient, $h_0 \to h_{10}$.

For filter length N = 22, $\omega_p = 0.4\pi$, edge frequency of stop-band $\omega_s = 0.6\pi$. The normalized amplitude response for H_0, H_1 filters of analysis bank and amplitude of distortion function $T'(\omega)$ are plotted in Figs. 2(a) and 2(b), respectively. From Fig. 2(c), this represents attenuation characteristic of low-pass filter H_0. Fig. 2(d) represents the reconstruction error of QMF bank. Table 1 provides the filter characteristics.

$h_0 =$ -0.00161 $h_1 =$ -0.00475 $h_2 =$ 0.01330 $h_3 =$ 0.00104 $h_4 =$ -0.02797
$h_5 =$ 0.00940 $h_6 =$ 0.05150 $h_7 =$ -0.03441 $h_8 =$ -0.10013 $h_9 =$ 0.12490
$h_{10} =$ 0.46861.

4.2.2 Two-Channel QMF Bank for N = 32, $\omega_p = 0.4\pi$, $\omega_s = 0.6\pi$, with 16 Filter Coefficient, $h_0 \to h_{16}$.

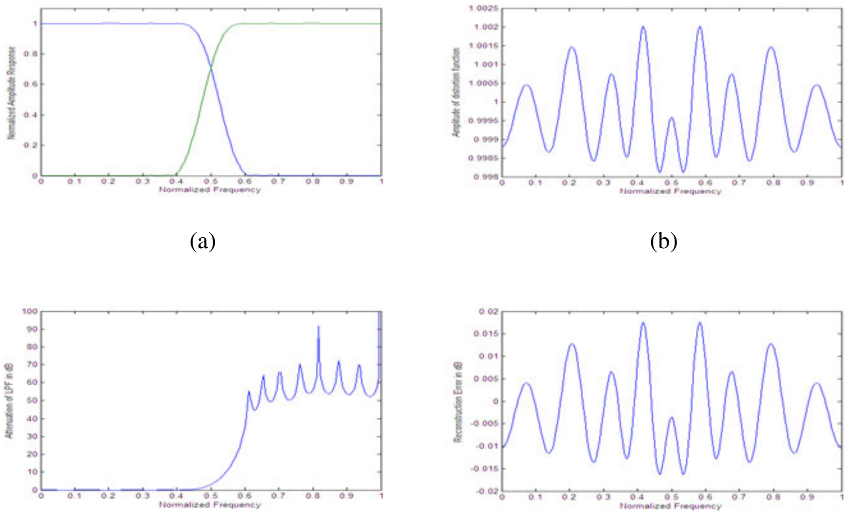
For filter length N = 32, $\omega_p = 0.4\pi$, edge frequency of stop-band $\omega_s = 0.6\pi$. After setting the initial values and parameters, the DEGL algorithm is run to obtain the optimal filter coefficients. The normalized amplitude response for H_0, H_1 filters of analysis bank and amplitude of distortion function $T'(\omega)$ are plotted in Figs. 4(a)



(a) (b)

(c) (d)

Fig. 2. The frequency response of Example 2. (a) Normalized amplitude response of analysis bank. (b) Amplitude of distortion function. (c) Low-pass filter attenuation characteristics in dB. (d) Reconstruction error in dB, for $N = 22$.



(a) (b)

(c) (d)

Fig. 3. The frequency response of Example 3. (a) Normalized amplitude response of analysis bank. (b) Amplitude of distortion function. (c) Low-pass filter attenuation characteristics in dB. (d) Reconstruction error in dB, for $N = 32$.

and 4(b), respectively. From Fig. 4(c), this represents attenuation characteristic of low-pass filter H_0. Fig. 4(d) represents the reconstruction error of QMF bank. Table 2 presents a comparison of proposed DEGL method with design based on CLPSO and DE algorithms. Table 1 provides the filter characteristics.

The optimized 16-filter coefficients for the analysis bank low-pass filter are given below.

$$h_0 = 0.0012; h_1 = \text{-}0.0024; h_2 = \text{-}0.0016; h_3 = 0.0057; h_4 = 0.0013; h_5 = \text{-}0.0115$$

$$h_6 = 0.0008;\ h_7 = 0.0201; h_8 = \text{-}0.0058; h_9 = \text{-}0.0330;\ h_{10} = 0.0167; h_{11} = 0.0539$$

$$h_{12} = \text{-}0.0417; h_{13} = \text{-}0.0997; h_{14} = 0.1306\ ; h_{15} = 0.4651$$

Tab1e 1. Filter's performance measuring quantities

	N=22	N=32
SBEA (stop-band edge attenuation)	20.8669 dB	34.7309 dB
SBLFA (stop-band first lobe attenuation)	31.1792 dB	44.6815 dB
MSEP(mean square error in pass band)	$8.95\times10^{=07}$	$1.58\times10^{=07}$
MSES (mean square error in stop band)	$1.23\times10^{=04}$	$3.15\times10^{=06}$
mor (measure of ripple)	0.0186	0.0083

Table 2. Performance comparison of proposed DEGL method with other algorithms

Name of the algorithm	Filter constants and parameters				
	mor	MSEP	MSES	SBEA	SBLFA
CL- PSO[19]	0.0256	$1.84\times10^{=04}$	$1.12\times10^{=03}$	23.6382	31.9996
DE[16]	0.0123	$9.72\times10^{=06}$	$6.96\times10^{=04}$	29.6892	39.8694
DEGL	0.0083	$1.58\times10^{=07}$	$3.15\times10^{=06}$	34.7309	44.6815

5 Discussion of Results

We have described here two design problems with no of filter coefficients equal to 11 and 16. One comparison table is also given for the 2nd design problem. In Table 2 the results for N=32 are compared with the results of same design problem using CL-PSO [19] and DE [16] algorithms. The results clearly show that this method leads to filter banks with improved performance in terms of peak reconstruction error, mean square error in stop band, pass band. The measure of ripple, which is an important parameter in signal processing, is also considerably lower than DE and CLPSO based methods for all the problems. Almost all the problems satisfy Eqn. 13 which is the condition for perfect reconstruction filter which is one of the achievements of this algorithm. The corresponding values of MSEP and MSES are also much lower in this method than the other two methods as shown in table 4. Also DEGL based method is much better in terms SBEA and SBLFA.

6 Conclusions

In this paper, a DEGL Algorithm based technique is used for the design of QMF bank Simulation The result shows that design of filter using DEGL is very effective and efficient for QMF filter design for any number of filter coefficients. We could also use other improved DE algorithms and many other evolutionary algorithms for design purpose. We can also use multi objective algorithms and multimodal optimization algorithms [21]. Our further works will be focused on the improvement of the result obtained using some new design scheme and further optimization techniques.

References

1. Vaidyanathan, P.P.: Multirate Systems and Filter Banks. Prentice Hall Inc. (1993)
2. Chandran, S., Ibrahim, M.K.: Adaptive antenna techniques using QMF bank. In: IEEE International Conference on Antennas and Propagation, pp. 257–260 (April 1995)
3. Petraglia, A., Mitra, S.K.: High speed A /D conversion incorporating a QMF. IEEE Trans. on Instrumentation and Measurement 41(3), 427–431 (1992)
4. Chan, S.C., Pun, C.K.S., Ho, K.L.: New design and realization techniques for a class of perfect reconstruction two-channel FIR filter banks and wavelet bases. IEEE Transactions Signal Processing 52(7), 2135–2141 (2004)
5. Johnston, J.D.: A filter family designed for use in quadrature mirror filter banks. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 291–294 (April 1980)
6. Pirani, G., Zingarelli, V.: An analytical formula for the design of quadrature mirror filters. IEEE Transactions Acoustic, Speech, Signal Processing ASSP-32, 645–648 (1984)
7. Jain, V.K., Crochiere, R.E.: Quadrature mirror filter design in time domain. IEEE Transactions on Acoustics Speech and Signal Processing ASSP-32, 353–361 (1984)
8. Chen, C.K., Leem, J.H.: Design of quadrature mirror filters with linear phase in the frequency domain. IEEE Transactions on Circuits and Systems II 39(9), 593–605 (1992)
9. Lim, Y.C., Yang, R.H., Koh, S.N.: The design of weighted minimax quadrature mirror filters. IEEE Transactions Signal Processing 41(5), 1780–1789 (1993)
10. Andrew, L., Franques, V.T., Jain, V.K.: Eigen design of quadrature mirror fil-ters. IEEE Trans. Circuits Syst. II Analog Digit.Signal Process. 44(9), 754–757 (1997)
11. 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)
12. Bregovic, R., Saramaki, T.: Two-channel FIR filter banks - A tutorial review and new results. In: Proceeding Second International Workshop on Transforms and Filter Banks, Brandenburg, Germany (March 1999)
13. Bregovic, R., Saramaki, T.: A general-purpose optimization approach for design-ing two-channel FIR filter banks. IEEE Transactions on Signal Processing 51(7) (July 2003)
14. Kumar, A., Singh, G.K., Anand, R.S.: Design of Quadrature Mirror Filter Bank us-ing Particle Swarm Optimization (PSO). International Journal of Recent Trends in Engineering 1(3) (May 2009)
15. Upendar, J., Gupta, C.P., Singh, G.K.: Design of two-channel quadrature mirror filter bank using particle swarm optimization. Elsevier Science Direct Digital Signal Processing 20, 304–313 (2010)
16. Storn, R., Price, K.: Differential evolution a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optimization 11(4), 341–359 (1997)

17. Mendes, R., Kennedy, J.: The fully informed particle swarm: Simpler may be better. IEEE Trans. Evol. Comput. 8(3), 204–210 (2004)
18. 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`
19. Liang, J.J., Qin, A.K., Suganthan, P.N., Baskar, S.: Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE Trans. Evol. Comput. 10(3), 281–295 (2006)
20. Das, S., Abraham, A., Chakraborty, U.K., Konar, A.: Differential Evolution Using a Neighborhood-Based Mutation Operator. IEEE Transactions on Evolutionary Computation 13(3), 526–553 (2009)
21. Qu, B.Y., Suganthan, P.N., Liang, J.J.: Differential Evolution with Neighborhood Mutation for Multimodal Optimization. IEEE Trans. on Evolutionary Computation, doi:10.1109/TEVC.2011.2161873

Differential Evolution with Modified Mutation Strategy for Solving Global Optimization Problems

Pravesh Kumar[1], Millie Pant[1], and V.P. Singh[2]

[1] Indian Insitute of Technology, Roorkee, India
[2] Millenium Insitute of Engineering and Technolgy, India
praveshtomariitr@gmail.com,
millifpt@iitr.ernet.in

Abstract. In the present work we propose a modified variant of Differential Evolution (DE) algorithm named MDE. MDE differs from the basic DE in the manner in which the base vector is generated. While in simple/basic DE, base vector is usually randomly selected from the population of individuals, in MDE base vector is generated as convex linear combination (clc) of three randomly selected vectors out of which one is the one having best fitness value. This mutation scheme is used stochastically with mutation scheme in which the base generated using a clc of three randomly generated vectors. MDE is validated on a set of benchmark problems and is compared with basic DE and other DE variants. Numerical and statistical analysis shows the competence of proposed MDE.

Keywords: differential evolution, mutation strategy, global optimization.

1 Introduction

Differential Evolution (DE) algorithm is an evolutionary algorithm, which was proposed by Storn and Price in 1995 [1]. Differential evolution (DE) [1], [2], is a simple and efficient, stochastic, population set based methods for global optimization over continuous space. DE is capable of handling non-differentiable, nonlinear and multimodal objective functions and have been applied successfully to a wide range of problem such as real-world applications, such as data mining [3], [4], pattern recognition, digital filter design, neural network training, etc. [5], Compared with Particle Swarm Optimization (PSO) and Genetic Algorithm (GA), DE algorithm has many advantages, such as faster convergence speed, stronger stability, easy to realize [6] and so on, so it is noticed by many researchers. In order to improve the performance of DE, its several variants have been proposed. There are various mutation strategies available in the literature [7]-[11]. In this paper we have taken the basic DE strategy *DE/rand/1/bin* [8], [9]. We shall refer to it as basic/ simple DE (SDE). In the present study we have proposed a new variant of DE, named MDE using a modified mutation strategy, based on [10].

The rest of paper is organized as follows: Section 2 provides a compact overview of basic DE. Section 3 presents the proposed MDE algorithm. Numerical Simulation and Comparisons are reported in Section 4, and finally the conclusions derived from the present study are drawn in Section 5.

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 11–18, 2011.
© Springer-Verlag Berlin Heidelberg 2011

2 Basic DE

Basic DE algorithm is a kind of evolutionary algorithm, used for function optimization [1]. The structure of DE is similar to the GA and both use the operators' *selection, mutation and crossover* to guide the search process. The main difference between standard GA and DE is mutation operation. Mutation is a main operation of DE, and it revises each individual's value according to the difference vectors of the population. The algorithm uses mutation operation as a search mechanism; crossover operation to induce diversity and selection operation to direct the search toward the potential regions in the search space.

The working of DE is as follows: First, all individuals are initialized with uniformly distributed random numbers and evaluated using the fitness function provided. Then the following are executed until maximum number of generation has been reached or an optimum solution is found.

Mutation: For a *D*-dimensional search space, for each target vector $X_{i,G}$ at the generation *G*, its associated mutant vector is generated via certain mutation strategy. The most often used mutation strategy implemented in the DE is given by equation-1.

$$DE/rand/1/bin\text{:}\ V_{i,G+1} = X_{r_1,G} + F*(X_{r_2,G} - X_{r_3,G}) \tag{1}$$

where $r_1, r_2, r_3 \in \{1,2,....,NP\}$ are randomly chosen integers, different from each other and also different from the running index *i.* F (>0) is a scaling factor which controls the amplification of the difference vectors.

Crossover: Once the mutation phase is over, crossover is performed between the target vector and the mutated vector to generate a trial point for the next generation.

The mutated individual, $V_{i,G+1} = (v_{1,i,G+1}, \ldots, v_{D,i,G+1})$, and the current population member (target vector), $X_{i,G} = (x_{1,i,G}, \ldots, x_{D,i,G})$, are then subject to the crossover operation, that finally generates the population of candidates, or "trial" vectors, $U_{i,G+1} = (u_{1,i,G+1}, \ldots, u_{D,i,G+1})$, as follows:

$$u_{j,i,G+1} = \begin{cases} v_{j,i,G+1} & if \quad rand_j \le Cr \vee j = k \\ x_{j,i,G} & otherwise \end{cases} \tag{2}$$

where $j, k \in \{1,\ldots, D\}$ *k* is a random parameter index, chosen once for each *i*, Cr is the crossover probability parameter whose value is generally taken as $Cr \in [0,1]$.

Selection: The final step in the DE algorithm is the selection process. Each individual of the temporary (trial) population is compared with the corresponding target vector in the current population. The one with the lower objective function value survives the tournament selection and goes to the next generation. As a result, all the individuals of the next generation are as good as or better than their counterparts in the current generation.

$$X_{i,G+1} = \begin{cases} U_{i,G+1} & \text{if} \quad f(U_{i,G+1}) \le f(X_{i,G}) \\ X_{i,G} & \text{otherwise} \end{cases} \tag{3}$$

3 Proposed Modified DE (MDE)

In this section, we describe the proposed *MDE*. Here, we used two new mutation strategies: first strategy (say *S1*) is selected from [10] and the second strategy (say *S2*) is novel strategy.

Both *S1* and *S2*, differ from the basic DE mutation strategy in the selection of base vector. While in basic DE, a single vector is selected as a base vector in *S1* and *S2*, the base vector is a convex linear combination of three randomly selected vectors of the population.

In S1, the three vectors are randomly selected while in *S2*, only two vectors are randomly selected and the third vector is the one having best fitness value $X_{best,\,G}$

The strategies, *S1* and *S2*, are defined as

$$\begin{aligned} S1: V_{i,\,G+1} &= (\mu_1 X_{r1,\,G} + \mu_2 X_{r2,\,G} + \mu_3 X_{r3,\,G}) + F(X_{r2,\,G} - X_{r3,\,G}) \\ S2: V_{i,\,G+1} &= (\mu_1 X_{best,\,G} + \mu_2 X_{r2,\,G} + \mu_3 X_{r3,\,G}) + F(X_{r2,\,G} - X_{r3,\,G}) \end{aligned} \tag{4}$$

Here μ_i $i=1, 2$ are uniform random number between 0 and 1 and $\mu_3=1-(\mu_1+\mu_2)$, (satisfies the condition $\sum_{i=1}^{3} \mu_i = 1$).

These strategies are stochastically applied using a fixed probability (*Pr*). A random number (*R*) is generated uniformly between 0 & 1. If the value of *R* is less than *Pr* then *S1* is selected otherwise is *S2* selected.

3.1 Pseudo Code for MDE

```
Begin
 Create uniformly random population $X_{i,G}$, where
 i=1,2,..,NP
   while(termination criteria is met)
     for i=1:NP
       {
         Select three random parents $X_{r1,G}$, $X_{r2,G}$ , $X_{r3,G}$
         and $X_{best,G}$ from the current population where
         $i \neq r1 \neq r2 \neq r3$

         {
           if(Pr<rand(0,1))
            Perform mutation S1 by using equation-4
           else
            Perform mutation S2 by using equation-4
         }
```

```
            Perform crossover and selection operation by
            using equation-2 and equation-3 respectively.
          }
     end while
  End
```

4 Numerical Simulation and Comparisons

4.1 Experimental Setting

In this section, the proposed MDE algorithm is validated on a set of eight benchmarks taken from [14]. In order to investigate the performance of the proposed MDE we compared it with the basic DE algorithm (SDE) and three other modified variants of DE; Trigonometric DE, (TDE)[12], SADE[13], and ISADE [14]. Basic DE, proposed MDE and TDE are implemented in Dev-C++ and the experiments are conducted on a computer with 2.00 GHz Intel (R) core (TM) 2 duo CPU and 2- GB of RAM. For SADE and ISADE, the results are taken from literature. For each problem, the MDE is independently run 30 times. The parameter setting is taken as follows:

Table 1. Parameter Setting

Pop size (*NP*)	100
Dimension (*D*)	30
Scale Factor (*F*), and Crossover rate (*Cr*)	0.5,0.9
Probability(*Pr*)	0,5
Value to reach (*VTR*)	10^{-08} except for f_4 where *VTR* 10^{-02}
Max NFE	300000

4.2 Performance Criteria

Four performance criteria are selected to evaluate the performance of the algorithms. These criteria are described as follows:

Mean Fitness and Standard Deviation: The average of function fitness value that an algorithm can find, using predefined maximum NFEs, is recorded in each run and then average of the function fitness values are calculated. Also the average and standard deviation of the fitness values are calculated.

NFEs [15]**:** The number of fitness function evaluations (NFEs) is recorded when the VTR is reached before to reach maximum NFE. i.e we set the termination criteria as $| f_{opt} - f_{globqal} | \leq VTR$ and record average NFE of successful run over 30 runs.

Convergence Graphs [15]**:** The convergence graphs show the mean fitness performance of the total runs, in the respective experiments.

Acceleration rate (AR) in %[16]: This criterion is used to compare the convergence speeds between MDE and other algorithms. It is defined as follows: $AR = \frac{NFE_{Others} - NFE_{MDE}}{NFE_{others}} \%$,

4.3 Simulated Results

The simulated results based on the above experimental setting are given in Table-2 and Table-3. In Table-2 we have taken the results on the basis of average fitness. Here, first we fixed the NFE at 10^5 and then take the average of minimum fitness function value in 30 runs. From the Table-2 we can clearly see that for all benchmark functions MDE gives better results than TDE and SDE except function f_3.In the case of f_3 we cannot reach minimum value than 0. A two sample *t-test* [17],[18] is also applied to analyze the statistical significance of the proposed algorithm. We have checked the significant difference of MDE with respect to DE and TDE at 5% level of significance. For statistical analysis, we consider the null hypothesis and alternative hypothesis as:

***H_0*: There is no significant difference between the mean fitness value of MDE and others algorithms.

***H_1*: Mean fitness value of MDE is better than others algorithms.

The calculated *t-value* of all function is greater than *t-table* value that shows the significant better performance in the comparison of SDE and TDE.

In the Table-3, we fixed *VTR* as given in experimental setting and then calculated the average NFE of 30 runs. From Table-3 we can see that the proposed MDE gives the better results for every function in the comparison to the other algorithms. From

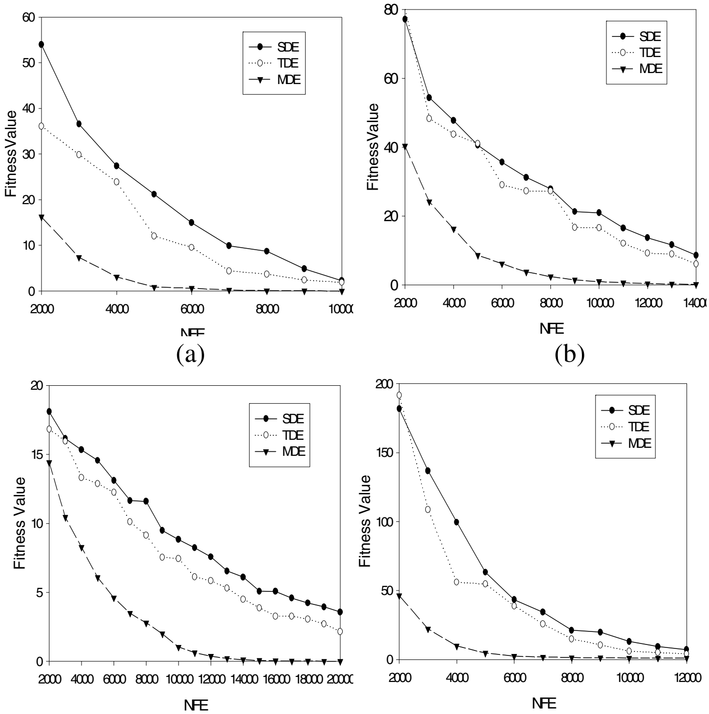
Table 2. Average function fitness value and standard deviation of 30 runs

Fun	SDE	TDE	MDE	*t-value* MDE/SDE	*t-value* MDE/TDE
f1	2.32e-012 ±6.45e-013	3.72e-015 ±1.98e-015	**3.71E-035** **±4.50E-035**	11.73	5.94
f2	2.26e-006 ±3.67e-007	8.20e-008 ±2.02e-008	**3.21e-019** **±7.84e-020**	19.47	12.83
f3	0±0	0±0	**0±0**	--	--
f4	0.01996 ±0.00251	0.01187 ±0.00301	**0.00328** **±0.00041**	20.73	8.94
f5	-7826.02 ±265.026	-7671.99 ±360.164	**-12569.4** **±134.52**	50.46	40.28
f6	16.684 ±1.4569	20.6849 ±2.62868	**0±0**	36.21	24.88
f7	7.53e-006 ±1.49e-006	2.03e-007 ±3.99e-008	**3.41e-015 ±0**	15.98	16.08
f8	1.29e-008 ±1.79e-008	4.45e-012 ±4.31e-012	**3.25e-020** **±4.96e-020**	2.27	3.36

Table 3. Comparison of MDE with other algorithms in terms of NFEs and AR(%), Here 5/2 implies MDE vs TDE, 5/3 implies MDE vs SADE and 5/4 implies MDE vs ISADE

F	SDE	TDE	SADE[14]	ISADE[14]	MDE	AR 5/2	AR 5/3	AR 5/4
f1	6.49E+04	5.42E+04	5.43E+04	4.90E+04	**2.43E+04**	55.16	55.24	50.40
f2	1.18E+05	9.91E+04	7.56E+04	6.46E+04	**4.35E+04**	56.10	42.46	32.66
f3	2.92E+04	2.38E+04	2.06E+04	1.89E+04	**1.04E+04**	56.30	49.51	44.97
f4	1.85E+05	1.35E+05	1.11E+05	1.02E+05	**3.44E+04**	74.51	69.00	66.27
f5	2.82E+05	2.89E+05	8.94E+04	7.77E+04	**5.87E+04**	79.68	34.34	24.45
f6	2.94E+05	NA	1.14E+05	8.62E+04	**4.87E+04**	NA	57.28	43.50
f7	1.27E+05	1.02E+05	8.26E+04	7.36E+04	**4.64E+04**	54.5	43.82	36.95
f8	9.07E+04	7.29E+04	5.74E+04	5.13E+04	**3.25E+04**	55.41	43.68	36.64
Average AR(%)						61.66	49.41	41.98

NA- when maximum NFE reached before VTR



(a) (b)

(c) (d)

Fig. 1. (a): Convergent graph of function *f1,* (b): Convergent graph of function *f2,* (c): Convergent graph of function *f7,* (d): Convergent graph of function *f8*

the Table-3 it is clear that the proposed MDE is faster than TDE by 61.66%, faster than SADE by 49.41% and faster than ISADE by 41.98%.

Figure1(a)-1(d) shows the convergence graph of MDE with the comparisons of TDE and SDE.

5 Conclusions

In the present study, we propose a new mutation scheme for DE algorithm, where the base vector is generated as convex linear combination of three randomly selected vectors out of which one is having the best fitness value. This scheme is applied stochastically with the scheme proposed in [10]. The corresponding algorithm called MDE is validated on a set 8 benchmark problems taken from literature and the results are compared with SDE as well as with some other modified variants of DE available in literature. It is shown that the proposed scheme in MDE though simple achieves quality results within a reasonably good convergence rate.

References

1. Storn, R., Price, K.: Differential Evolution—A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces. Berkeley, CA, Tech. Rep. TR-95-012 (1995)
2. 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)
3. Alatas, B., Akin, E., Karci, A.: Modenar: Multi-Objective Differential Evolution Algorithm for Mining Numeric Association Rules. Applied Soft Computing 8(1), 646–656 (2008)
4. Das, S., Abraham, A., Konar, A.: Automatic Clustering Using an Improved Differential Evolution Algorithm. IEEE Transaction on Systems Man and Cybernetics: Part A 38(1), 218–237 (2008)
5. Price, K., Storn, R., Lampinen, J.: Differential Evolution: A Practical Approach to Global Optimization. Springer, Berlin (2005)
6. Vesterstrom, J., Thomsen, R.: A Comparative Study of Differential Evolution, Particle Swarm Optimization and Evolutionary Algorithms on Numerical Benchmark Problems. In: Congress on Evolutionary Computation, pp. 980–987 (2004)
7. Montes, E.M., Reyes, J.V.: A Comperative Study of Differential Evolution Variants for Global Optimization. In: GECCO, Seattle Washington USA, pp. 485–492 (2006)
8. Pant, M., Ali, M., Abraham, A.: Mixed Mutation Strategy Embedded Differential Evolution. In: IEEE Congress on Evolutionary Computation, pp. 1240–1246 (2009)
9. Noman, N., Iba, H.: Accelerating Differential Evolution Using an Adaptive Local Search. IEEE Transactions on Evolutionary Computation 12, 107–125 (2008)
10. Fan, H.Y., Lampinen J., Dulikravich, G.S.: Improvements to Mutation Donor Formulation of Differential Evolution. In: International Congress on Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems Eurogen (2003)
11. Ali, M.M.: Differential Evolution with Preferential Crossover. European Journal of Operational Research 181, 1137–1147 (2007)
12. Fan, H., Lampinen, J.: A Trigonometric Mutation Operation to Differentia Evolution. Journal of Global Optimization 27, 105–112 (2003)
13. Brest, J., Greiner, S., Bošković, B., Mernik, M., Žumer, V.: Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems. IEEE Transactions on Evolutionary Computation 10(6), 646–657 (2006)

14. Jia, L., Gong, W., Wu, H.: An Improved Self-adaptive Control Parameter of Differential Evolution for Global Optimization. In: Cai, Z., Li, Z., Kang, Z., Liu, Y. (eds.) ISICA 2009. CCIS, vol. 51, pp. 215–224. Springer, Heidelberg (2009)
15. Suganthan, P., Hansen, N., Liang, J.: Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization (2005)
16. Rahnamayan, S., Tizhoosh, H., Salama, M.: Opposition-Based Differential Evolution. IEEE Transactions on Evolutionary Computation 12(1), 64–79 (2008)
17. Zhu, R.: Statistical Analysis Methods. China Forestry Publishing House, Beijing (1989)
18. Zhang, M., Luo, W., Wang, X.: Differential Evolution with Dynamic Stochastic Selection for Constrained Optimization. Information Science: An International Journal 178, 3043–3074 (2008)

Self-adaptive Cluster-Based Differential Evolution with an External Archive for Dynamic Optimization Problems

Udit Halder[1], Dipankar Maity[1], Preetam Dasgupta[1], and Swagatam Das[2]

[1] Dept. of Electronics & Tele-comm. Engineering, Jadavpur University, Kolkata
[2] Electronics and Comm. Sciences Unit, Indian Statistical Institute, Kolkata
udithalder99@gmail.com,
dipankarmaity1991@gmail.com,
dasguptapreetam@gmail.com, swagatam.das@ieee.org

Abstract. In this paper we propose a self adaptive cluster based Differential Evolution (DE) algorithm to solve the Dynamic Optimization Problems (DOPs). We have enhanced the classical DE to perform better in dynamic environments by a powerful clustering technique. During evolution, the information gained by the particles of different clusters is exchanged by a self adaptive strategy. The information exchange is done by re-clustering, and the cluster number is updated adaptively throughout the optimization process. To detect the environment change a test particle is used. Moreover, to adapt the population in new environment an External Archive is also used. The performance of SACDEEA is evaluated on GDBG benchmark problems and compared with other existing algorithms.

Keywords: Differential Evolution, Dynamic Optimization Problem (DOP), Self adaptive clustering.

1 Introduction

Dynamic Optimization Problems (DOPs) are those optimization problems whose optimal solution changes over time during the optimization. Such dynamic optimization problems (DOPs) can be defined as follows:

$$F = f(x, \phi, t); \tag{1}$$

where F is the optimization problem, f is the cost function, x is a feasible solution in the solution set X, t is time, and ϕ is the system control parameter. In static optimization problem, the optima do not change their position and height, but in dynamic problem they are functions of time; i.e. the position and height of optima changes over time. A good dynamic optimization algorithm should not only be able to find the global optimum but also detect whether an environment change has occurred or not, and if yes then it should track the new optimum in the changed environment.

In recent years, solving DOPs with Evolutionary Algorithms (EAs) has attracted much interest among researchers. However, most of the analysis and research were done on EAs in static optimization problems. So, properly enhanced EAs can tackle DOPs. An early approach of using Genetic Algorithm (GA) for DOPs could be found

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 19–26, 2011.
© Springer-Verlag Berlin Heidelberg 2011

in [3]. Differential Evolution (DE), [1, 10-13], is a powerful evolutionary algorithm for solving real-parameter optimization problems. DE has also been used to address DOPs [4]. Mendes and Mohais proposed DynDE [5], a multi-population DE algorithm for efficiently solving the Moving Peaks Benchmark (MPB) described by Branke [6]. Brest *et al.* [8] investigated a self-adaptive DE algorithm (*jDE*) where the control parameters *F* and *CR* are self adapted and multi population was used. Yang and Li [7] proposed a Clustering Particle Swarm Optimizer (CPSO) for DOPs. A Particle Swarm Optimizer with Composite Particles (PSO-CP) is proposed by Liu, Yang and Wang [9].

In this paper, we propose a self adaptive cluster based DE-variant with an External Archive to solve the DOPs. The total population is divided in several groups and optimization is done separately in each group. During the optimization process, redistribution takes place self adaptively. Also, an external archive is used to perform well in dynamic environment.

2 Differential Evolution (DE) Algorithm

Differential Evolution (DE) is a very simple but a very powerful algorithm for optimization problem. Let $S \in R^n$ be the search space of the problem under consideration. DE algorithm starts with an initial population of *NP, n* dimensional solution particles. The particles are of the form $\vec{X}_i = (x_{1,i}, x_{2,i}, \ldots, x_{n,i}) \in S$, where *i=1, 2, ..., NP* and are upgraded from one generation to next generation, where $x_{i1}, x_{i2}, \ldots., x_{in}$ are in between their respective upper and lower bounds x_j^{upper}, x_j^{lower} respectively.

2.1 Mutation

After initialization for each solution vector $\vec{X}_{i,t}$, a new vector $\vec{Y}_{i,t}$ is generated at each generation *t*. The vector $\vec{Y}_{i,t}$ can be generated by any one of the five methods; DE/rand/1, DE/best/1, DE/target-to-best/1, DE/best/2, DE/rand/2. We used the DE/best/1 scheme for our algorithm as given by:

$$\vec{Y}_{i,t} = \vec{X}_{best,t} + F \cdot (\vec{X}_{r_1^i,t} - \vec{X}_{r_2^i,t}) \tag{2}$$

2.2 Cross-Over

After the phase of mutation the crossover phase plays a major role to enhance the diversity of the population. In this phase the generated vector $\vec{Y}_{i,t}$ exchanges its component with its parent vector $\vec{X}_{i,t}$ to generate a new vector $\vec{U}_{i,t} = (u_{1,i,t}, u_{2,i,t}, \ldots, u_{n,i,t})$, where $u_{j,i,t}$ is found by the following procedure:

$$u_{j,i,t} = \begin{cases} y_{j,i,t}\ , & \text{if } rand_{i,j}(0,1) < Cr \text{ or } j = j_{rand} \\ x_{j,i,t}\ , & \text{otherwise} \end{cases}$$

where $rand_{i,j}(0,1)$ is a uniformly distributed random number in the range 0 and 1. $j_{rand} \in [1, 2, ..., n]$ is a randomly chosen index, which ensures that $\vec{U}_{i,t}$ gets at least one component from $\vec{Y}_{i,t}$, and *Cr* is a user defined constant in the range (0,1).

2.3 Selection Operation

In this phase a greedy selection is made between the target vector and generated vector for next generation. The selection procedure is done by the following way:

$$\vec{X}_{i,t+1} = \begin{cases} \vec{U}_{i,t} & \text{if } f(\vec{U}_{i,t}) \le f(\vec{X}_{i,t}) \\ \vec{X}_{i,t} & \text{if } f(\vec{U}_{i,t}) > f(\vec{X}_{i,t}) \end{cases}$$

for minimization problem. Thus after every generation we find either a new solution which has better fitness (here for minimization problem) or the previous vector is kept. So after each generation the population gets better or remains unchanged but never deteriorates.

3 Proposed Algorithm: SACDEEA

In this paper we propose a strategy based, self-adaptive, clustered DE with an External Archive to solve the DOPs. Here, in this algorithm we divide the whole population in several clusters based on their spatial positions. Before describing the algorithm in details let us define some terms regarding the clusters.

- If the particles of a cluster are $\vec{x}_1, \vec{x}_2 ....., \vec{x}_m$ then the center ($\vec{C}$) is determined as:

$$\vec{C} = \frac{\sum_{i=1}^{m} \vec{x}_i}{m} \tag{3}$$

- The radius (*R*) of a cluster is defined as the mean distance (Euclidean) of the particles from the center of the cluster. So, we can write

$$R = \frac{\sum_{i=1}^{m} \sqrt{\sum_{j=1}^{n} (x_{i,j} - C_j)^2}}{m}; \tag{4}$$

where n is the dimension of the problem.

- For any cluster, we define a fixed value, named *convergence radius* (R_{conv}), which is calculated as

$$R_{conv} = dist\,(\vec{X}_{upper}, \vec{X}_{lower}) \cdot 10^{-3} \quad (5)$$

where $\vec{X}_{upper}$ and $\vec{X}_{lower}$ are the upper bound and lower bound vectors in the search region, and $dist(\vec{a}, \vec{b})$ denotes the Euclidean distance between the vectors $\vec{a}$ and $\vec{b}$.

- Every cluster can contain up to a maximum number (*pop_max*) of individuals.

It has been seen that DE performs better than other EAs in many optimization problems. So, we used DE as our main improvement algorithm with some enhancements to handle dynamic problems. The modifications we have made are described below.

3.1 Initialization

A certain number of individuals (NP) are spread randomly in the search region. Then they are clustered into k clusters using the K-means clustering algorithm.

3.2 Cluster Improvements

After clustering is done, evolutionary strategy is used to improve the members of the cluster. For this purpose, “DE/best/1” scheme (2) is used separately in each cluster, i.e. there is no information sharing among the clusters in a specific generation. This might prevent the algorithm to converge all the particles to a single local optimum or in other words, premature convergence is discouraged. The clustering technique is used to explore the search region more and search for better and promising areas simultaneously instead of optimizing a single population and therefore guiding all the particles to converge to one optimum, which may be a false one. By this way, the diversity of the population can be maintained.

3.3 Performance Evaluation and Redistribution

During optimization, it may so happen that two or more clusters get overlapped and therefore searching in a same region. So, redistribution is necessary over time. This is implemented in our algorithm with an extra technique of performance evaluation. After a *time span* (*TS*) or generations the performance of the algorithm is evaluated and accordingly the cluster numbers are updated. Performance of the algorithm is evaluated in terms of number of changes in global best value over a *TS*. If the change is satisfactory i.e. if number of changes is greater than a fixed value *change_ref*, then cluster number is reduced, otherwise it is increased. However, the number of clusters remains between two limits called *max_cluster* & *min_cluster*, which are the maximum and minimum allowed cluster numbers. After updating the number of clusters, the total population is re-clustered in updated cluster numbers. When the algorithm is performing well, it reduces the cluster number. However, when the performance evaluation test gives bad results, then the cluster number is increased by one. A fresh cluster is injected in the population, whose individuals are generated randomly. This process can help us to maintain the diversity of the population, which is an important aspect regarding to DOPs.

3.4 Detection of Environmental Change and External Archive

It was stated that an efficient algorithm should detect when the environment changes. For the detection purpose, we place a particle, called *test particle*, in the search space. The *test particle* does not take part in the optimization process; rather its objective value is constantly evaluated in each generation. If its new function value does not match with the previous value, then we can say that an environmental change has occurred. Then we stop continuing the optimization process and it is restarted afresh, i.e. the algorithm starts from the Initialization step.

An algorithm can be stated efficient in dynamic environments if it can utilize the knowledge gained in the previous environment on the changed one. For this purpose, we have used an *External Archive*. During optimization, the radius of each cluster is calculated. If the radius becomes smaller than R_{conv}, described in (5), then we can say that the cluster is highly converged around a point. So then the position of the best particle of that cluster is stored in an Archive and the cluster is deleted. When a change in environment is detected, then the particles of the Archive, which were the optima in previous environment, are added to the new population, expecting they could help in tracking the movement of optima. If the change in environment is not too severe, this method should improve the search in new environment.

4 Experimental Results

4.1 Test Problems

In this paper, our algorithm is tested on the GBDG benchmark problems [2]. The test suite has 6 problems F1-F6. The number of peaks of F1 is 10, 50 and other functions have 10 peaks each. For our simulation purpose, we took the change instances T1 to T6. The performance of our algorithm is compared with other 3 significant dynamic optimizers. The functions are evaluated over 60 environment changes in a single run and 20 such runs are recorded. The mean and standard deviation values of those runs are shown in tables 1-7.

4.2 Parameter Settings

NP = 200; $F = 0.5*(rand+1)$; Cr = 0.9; k = 5; $max\_cluster = 2k = 10$, $min\_cluster$ = 1; pop_max = 100; TS = 10; R_{conv} can be calculated from (5).

Table 1. Error values achieved for F1 with number of peaks = 10

Algorithm		Change instance					
		T1	T2	T3	T4	T5	T6
SACDEEA	Mean Std.	**1.32e-05** **4.67e-05**	**1.23e+00** **5.54e+00**	1.32e+01 3.20e+01	**9.67e-04** **2.76e-03**	**7.23e-01** **2.02e+00**	**1.02e+00** **3.90e+00**
jDE	Mean Std.	2.88e-02 4.43e-01	3.59e+00 7.84e+00	3.00e+01 7.13e+01	1.53e-02 2.88e-01	2.18e+00 4.39e+00	1.15e+00 5.73e+00
CPSO	Mean Std.	3.51e-02 4.26e-01	2.72e+00 6.52e+00	**4.13e+00** **9.00e+00**	9.44e-02 7.86e-01	1.87e+00 4.49e+00	1.06e+00 4.81e+00
DynDE	Mean Std.	7.32e-02 2.96e+00	2.55e+00 8.43e+00	5.42e+00 9.24e+00	1.26e-01 9.42e-01	1.56e+00 4.64e+00	1.31e+00 6.25e+00

Table 2. Error values achieved for F1 with number of peaks = 50

Algorithm		Change instance					
		T1	T2	T3	T4	T5	T6
SACD EEA	Mean	**1.32e-03**	**1.34e+00**	4.83e+00	**3.87e-04**	**5.98e-03**	**5.73e-02**
	Std.	**3.89e-02**	**5.96e+00**	7.29e+00	**1.89e-02**	**4.67e-01**	**2.03e+00**
jDE	Mean	1.72e+00	4.09e+00	**4.29e+00**	8.77e-02	9.48e-01	1.77e+00
	Std.	7.64e-01	6.45e+00	**6.75e+00**	2.46e-01	1.77e+01	5.83e+00
CPSO	Mean	2.63e-01	3.28e+00	6.32e+00	1.25e-01	8.48e-01	1.48e+00
	Std.	9.36e-01	5.30e+00	7.44e+00	3.86e-01	1.78e+00	4.39e+00
DynDE	Mean	3.29e-01	4.65e+00	6.46e+00	1.41e-01	1.02e+00	9.86e-01
	Std.	1.52e+00	6.34e+00	9.35e+00	5.91e-01	2.65e+00	4.86e+00

Table 3. Error values achieved for F2

Algorithm		Change instance					
		T1	T2	T3	T4	T5	T6
SACDEEA	Mean	**3.09e-02**	**6.22e+00**	**4.54e+00**	**9.34e-02**	**1.76e+01**	**9.83e-01**
	Std.	**2.09e-01**	**3.12e+00**	**3.45e+00**	**3.56e-01**	**4.32e+01**	**2.90e+00**
jDE	Mean	9.63e-01	4.30e+01	5.02e+01	7.93e-01	6.71e+01	3.37e+00
	Std.	3.08e+01	1.15e+02	1.24e+02	2.53e+00	1.30e+02	1.28e+01
CPSO	Mean	1.25e+00	1.01e+01	1.03e+01	5.67e-01	2.51e+01	1.99e+00
	Std.	4.18e+00	3.51e+01	3.35e+01	2.14e+00	6.43e+01	5.22e+00
DynDE	Mean	1.36e+00	1.30e+01	1.19e+01	7.84e-01	2.08e+01	2.18e+00
	Std.	5.03e+00	4.82e+01	4.57e+01	2.22e+00	6.45e+01	3.96e+00

Table 4. Error values achieved for F3

Algorithm		Change instance					
		T1	T2	T3	T4	T5	T6
SACDEEA	Mean	**1.07e+01**	**4.95e+02**	**5.80e+02**	**5.86e+01**	6.01e+02	**2.30e+02**
	Std.	**3.23e+01**	**2.04e+02**	**1.52e+02**	**1.79e+02**	2.95e+02	**4.00e+02**
jDE	Mean	1.14e+01	5.58e+02	5.72e+02	6.57e+01	**4.76e+02**	2.43e+02
	Std.	5.81e+01	3.85e+02	3.86e+02	2.09e+02	**3.80e+02**	3.85e+02
CPSO	Mean	1.38e+02	8.55e+02	7.66e+02	4.31e+02	8.60e+02	7.53e+02
	Std.	2.22e+02	1.61e+02	2.36e+02	4.32e+02	1.22e+02	3.62e+02
DynDE	Mean	2.12e+01	7.92e+02	6.36e+02	3.42e+02	7.49e+02	5.19e+02
	Std.	7.37e+01	2.56e+02	3.43e+02	4.19e+02	2.81e+02	4.38e+02

Table 5. Error values achieved for F4

Algorithm		Change instance					
		T1	T2	T3	T4	T5	T6
SACDEEA	Mean	**1.24e+00**	**2.43e+00**	**1.34e+01**	2.97e+00	1.20e+02	**7.67e-01**
	Std.	**5.46e+00**	**5.67e+00**	**5.90e+01**	2.63e+00	1.93e+01	**1.09e+01**
jDE	Mean	1.49e+00	4.95e+00	5.19e+01	1.51e+00	6.94e+01	2.35e+00
	Std.	4.48e+00	1.35e+00	1.42e+02	4.10e+00	1.44e+00	5.78e+00
CPSO	Mean	2.68e+01	3.72e+01	3.67e+01	**7.93e-01**	6.72e+01	4.88e+00
	Std.	7.06e+00	9.94e+01	9.72e+01	**2.78e+00**	1.30e+02	1.54e+01
DynDE	Mean	1.86e+00	3.95e+01	2.35e+01	8.69e-01	**4.47e+01**	1.56e+00
	Std.	5.75e+00	9.86e+01	9.45e+01	3.17e+00	**1.22e+02**	6.21e+00

Table 6. Error values achieved for F5

Algorithm		Change instance					
		T1	T2	T3	T4	T5	T6
SACDEEA	Mean	**2.90e-02**	1.89e+00	**2.74e-01**	**3.09e-02**	**1.38e-01**	**1.23e-01**
	Std.	**3.09e+00**	2.89e+01	**1.98e+00**	**3.02e-01**	**2.90e+00**	**3.78e-01**
jDE	Mean	1.60e-01	**3.34e-01**	3.58e-01	1.08e-01	4.09e-01	2.30e-01
	Std.	1.03e+00	**1.64e+00**	1.83e+00	8.27e-01	1.91e+00	9.35e-01
CPSO	Mean	1.86e+00	2.88e+00	3.40e+00	1.10e+00	7.99e+00	4.05e+00
	Std.	5.18e+00	6.79e+00	6.45e+00	4.87e+00	1.38e+01	8.37e+00
DynDE	Mean	2.99e+00	2.94e+00	2.91e+00	1.38e+00	8.44e+00	2.30e+00
	Std.	6.88e+00	4.72e+00	5.39e+00	2.42e+00	1.21e+01	3.62e+00

Table 7. Error values achieved for F6

Algorithm		Change instance					
		T1	T2	T3	T4	T5	T6
SACDEEA	Mean	**4.78e+00**	2.78e+01	**5.98e+00**	**3.02e+00**	**1.05e+00**	**2.98e+00**
	Std.	**1.23e+01**	1.98e+01	**1.87e+01**	**1.78e+01**	**2.36e+01**	**1.13e+00**
jDE	Mean	6.23e+00	**1.03e+01**	1.10e+01	6.79e+00	1.49e+01	7.80e+00
	Std.	1.04e+01	**1.32e+01**	2.33e+01	1.02e+01	4.52e+01	1.10e+01
CPSO	Mean	6.73e+00	2.16e+01	2.71e+01	9.27e+00	7.16e+01	2.37e+01
	Std.	9.97e+00	6.35e+01	8.40e+01	2.42e+01	1.60e+02	5.16e+01
DynDE	Mean	6.04e+00	2.02e+01	1.93e+01	8.87e+00	4.34e+01	1.21e+01
	Std.	1.10e+01	6.22e+01	6.74e+01	2.67e+01	1.37e+02	2.52e+01

5 Conclusion

This paper proposes a self adaptive cluster based Differential Evolution algorithm for solving DOPs. We have evaluated the performance of our algorithm on the benchmark problems of GDBG system. There are six test functions and the numerical results show satisfactory performance of the algorithm. However, we have to enhance our algorithm such that it can detect dimension change also. So, our future work will include this change instance. It will also be attempted to improve the self adaptive nature of the algorithm to get further good results.

References

1. 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)
2. Li, C., Yang, S., Nguyen, T.T., Yu, E.L., Yao, X., Jin, Y., Beyer, H.G., Suganthan, P.N.: Benchmark Generator for CEC 2009 Competition on Dynamic Optimization. University of Leicester, Univ. of Birmingham, Nanyang Technological University, Tech. Rep. (2008)
3. Grefenstette, J.J.: Genetic algorithms for changing environments. In: Proc. 2nd Int. Conf. Parallel Problem Solving from Nature, pp. 137–144 (1992)
4. Angira, R., Santosh, A.: Optimization of dynamic systems: A trigonometric differential evolution approach. Computers & Chemical Engineering 31(9), 1055–1063 (2007)
5. Mendes, R., Mohais, A.S.: DynDE: a differential evolution for dynamic optimization problems. In: Proc. of IEEE Cong. on Evol. Comput., vol. 2, pp. 2808–2815 (2005)
6. Branke, J.: Memory enhanced evolutionary algorithms for changing optimization problems. In: Proc. of IEEE Congress on Evolutionary Computation, vol. 3, pp. 1875–1882 (1999)
7. Yang, S., Li, C.: A Clustering Particle Swarm Optimizer for Locating and Tracking Multiple Optima in Dynamic Environments. IEEE Transactions on Evolutionary Computation 14, 959–974 (2010)
8. Brest, J., Zamuda, A., Boskovic, B., Maucec, M.S., Zumer, V.: Dynamic Optimization using Self-Adaptive Differential Evolution. In: Proc. Cong. on Evol. Comput., pp. 415–422 (2009)
9. Liu, L., Yang, S., Wang, D.: Particle Swarm Optimization with Composite Particles in Dynamic Environments. IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics 40(6) (December 2010)
10. Das, S., Suganthan, P.N.: Differential Evolution: A Survey of the State-of-the-art. IEEE Trans. on Evolutionary Computation 15(1), 4–31 (2011)
11. Mallipeddi, R., Suganthan, P.N., Pan, Q.K., Tasgetiren, M.F.: Differential evolution algorithm with ensemble of parameters and mutation strategies. Applied Soft Computing 11(2), 1679–1696 (2011)
12. Neri, F., Tirronen, V.: Recent advances in differential evolution: a survey and experimental analysis. Artif. Intell., Rev. 33(1-2), 61–106 (2010)
13. Mallipeddi, R., Suganthan, P.N.: Ensemble of Constraint Handling Techniques. IEEE Trans. on Evolutionary Computation 14(4), 561–579 (2010)

An Informative Differential Evolution with Self Adaptive Re-clustering Technique

Dipankar Maity, Udit Halder, and Preetam Dasgupta

Dept. of Electronics and Tele-communication Engineering, Jadavpur University,
Kolkata – 700032, India
{dipankarmaity1991,udithalder99,dasguptapreetam}@gmail.com

Abstract. We propose an informative Differential Evolution (DE) algorithm where the information gained by the individuals of a cluster will be exchanged after a certain number of iterations called refreshing gap. The DE is empowered with a clustering technique to improve its efficiency over multimodal landscapes. During evolution, self-adaptive behaviour helps in re-clustering. With the better explorative power of the proposed algorithm we have used a new local search technique for fine tuning near a suspected optimal position. The performance of the proposed algorithm is evaluated over 25 benchmark functions and compared with existing algorithms.

Keywords: Differential Evolution, optimization, cluster, self-adaptive re-clustering.

1 Introduction

Scientist and Engineers from all branches have to deal with the global optimization problem where the main target is to find globally maximum or minimum value for a specified objective or cost function. Traditional techniques like steepest decent, linear programming, fail to find the global optima for most of the cases. To solve these kind of problems many optimization algorithm have been invented, which are based on any natural behavior like Particle Swarm Optimization(PSO)[1] , that imitate bird flocking nature, Ant Colony Optimization (ACO)[2] is based on the foraging technique of ant, Genetic Algorithm(GA)[3] based on the principle of the Darwinian theory of the survival of the fittest and the theory of evolution of the living beings; and many others like Artificial Immune Algorithm(AIA)[4], Artificial Bee Colony(ABC) [5], Bacteria Foraging Optimization (BFO) [6], etc. Differential Evolution (DE) [7, 13-16] proposed by Storn and Price is a very simple but one of the most promising algorithm for global optimization problems and it is not motivated by biological models. Now researchers have also incorporated local search technique with GA to form Memetic Algorithm(MA), that also shows better efficiency in optimization. In many applications such as Pattern Recognition [7], Communication [8], mechanical engineering [9], Real world optimization the effectiveness and efficiency of DE and other algorithms has been successfully demonstrated. But most of the algorithms and classical DE also fails to find the global optimum (optima) when the problem is high dimensional, highly multimodal. So here we propose a new algorithm called Informative

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 27–34, 2011.
© Springer-Verlag Berlin Heidelberg 2011

Differential Evolution with Self Adaptive Re-clustering Technique (IDE-SR) to find the most promising solutions for such problems. The information exchange through the re-clustering technique is the main theme of this paper. The rest of the paper is organized as follows section 2 contains related research works on the problems, section 3 is about basic DE algorithm, section 4 is the description of our proposed algorithm, section 5 gives the set of parameters and section 6 gives the experimental results of our algorithm on benchmark problem set [10] and compare with other algorithms and section 7 concludes the paper.

2 Related Research Works

A lot of researches have been done for the improvement of DE and also there are a lot of research works which have tried to optimize the CEC-2005 benchmark functions. Some of them are:

- A. K. Qin, V. L. Huang, and P. N. Suganthan, proposed a self-adaptive nature for solving these problems [11].
- Anne Auger and Nikolaus Hansen proposed a CMA evolution strategy by tuning the population size [12].

3 Differential Evolution (DE) Algorithm

Differential Evolution (DE) is a very simple but a very powerful algorithm for optimization problem. Let $S \in R^n$ be the search space of the problem under consideration. DE algorithm starts with an initial population of *NP, n* dimensional solution particles. The particles are of the form $\vec{X}_i = (x_{1,i}, x_{2,i}, x_{3,i}, ..., x_{n,i}) \in S$, where *i=1,2,....,NP* and are upgraded from one generation to next generation, where $x_{i1}, x_{i2}, ..., x_{in}$ are in between their respective upper and lower bounds x_j^{upper}, x_j^{lower} respectively. The population undergoes through Crossover, Mutation at each generation *t* and produces a new solution vector $\vec{U}_{i,t}$ for each vector $\vec{X}_{i,t}$.

3.1 Mutation

After initialization for each solution vector $\vec{X}_{i,t}$, a new vector $\vec{V}_{i,t}$ is generated at each generation *t*. There are some methods to generate $\vec{V}_{i,t}$, DE/rand/1, DE/best/1, DE/target-to-best/1, DE/best/2, DE/rand/2 etc. are some most useful methods. We used the DE/best/1 for our algorithm.

$$\vec{V}_{i,t} = \vec{X}_{best,t} + F.(\vec{X}_{r_1^i,t} - \vec{X}_{r_2^i,t}), \tag{1}$$

where r_1^i, r_2^i are mutually exclusive integers randomly chosen in the range [1,*NP*] and both are different from the index *i*.

3.2 Cross-Over

After the phase of mutation the crossover phase plays a major role to enhance the diversity of the population. In this phase the generated vector $\vec{V}_{i,t}$ exchanges its component with its parent vector $\vec{X}_{i,t}$ to generate a new vector $\vec{U}_{i,t} = (u_{1,i,t}, u_{2,i,t}, u_{3,i,t}, \ldots, u_{n,i,t})$, where $u_{j,i,t}$ is found by the following procedure:

$$u_{j,i,t} = \begin{cases} v_{j,i,t} \ , & if \quad rand_{i,j}(0,1) < Cr \quad or \quad j = j_{rand} \\ x_{j,i,t} \ , & otherwise \end{cases}$$

where $rand_{i,j}(0,1)$ is a uniformly distributed random number in the range (0,1). $j_{rand} \in [1,2,\ldots,n]$ is a randomly chosen index, which ensures that $\vec{U}_{i,t}$ gets at least one component from $\vec{V}_{i,t}$, and Cr is a user defined constant in the range [0,1].

3.3 Selection Operation

In this phase a greedy selection is made between the target vector and generated vector for next generation. The selection procedure is done by the following way:

$$\vec{X}_{i,t+1} = \begin{cases} \vec{U}_{i,t} & if \ f(\vec{U}_{i,t}) \le f(\vec{X}_{i,t}) \\ \vec{X}_{i,t} & if \ f(\vec{U}_{i,t}) > f(\vec{X}_{i,t}) \end{cases}$$

for minimization problem. Thus after every generation we find either a new solution which has better fitness (here for minimization problem) or the previous vector is kept. So after each generation the population gets better or remains unchanged but never deteriorates.

4 Proposed Algorithm

In our proposed algorithm we have used an information exchange strategy. We have adopted a self-adaptive cluster based strategy with a novel Local Search technique. The total population is divided into some clusters using K-means clustering algorithm; this is done to avoid the trapping of particles in any local optimum i.e. if the population is not divided then they can prematurely converge to a single local optima, whereas if they are clustered then it is expected that even though the premature convergence occurs, but different cluster will be converged to different optima. Each cluster exchanges information only after a certain number of iterations. After those iterations - which will be called as *refreshing gap* (*g_r*) for the rest of the paper- the total population is again clustered, now the cluster number may change or remain same. By this re-clustering technique, particles that were in different clusters in previous *g_r* can belong to same cluster now. So there will be a chance among those particles to exchange information that they have learned in past *g_r*. This information exchange helps them to explore a better position than that could be found by not using this re-clustering technique. The cluster number is varied in a self adaptive manner, here we used that if the algorithm performs satisfactorily well then cluster number is decreased until the cluster number is *cluster_min*, otherwise it is increased until it is *cluster_max*. It may so happen that two or more clusters can come closer to each other while improving them, then in the next *g_r* when we will re-cluster the total population then these clusters will form a single group as the K-means algorithm generates clusters based on the spatial distribution of the particles. So this algorithm does not require further checking of the overlapping of two or more clusters. We used DE/best/1 to each cluster separately because this variation of DE has faster convergence. In multimodal landscapes when a cluster has converged to an optimum then all the particles in that cluster come very closer so the difference vector in equation (1) becomes very small. Due to this premature convergence the rest of the FEs are wasted. So to save FE and for the fine tuning of the suspected optimum we have used a Local Search technique. In each iteration we calculated the radius of the clusters by the following rule,

$$R = \frac{\sum_{i=1}^{m} \sqrt{(\vec{x}_i - \vec{c})^T . (\vec{x}_i - \vec{c})}}{m}, \tag{2}$$

where m is the number of particles and $\vec{c}$ is the centre of the cluster defined as

$$\vec{c} = \frac{\sum_{i=1}^{m} \vec{x}_i}{m}. \tag{3}$$

If R is less than a pre-specified value *R_limit,* then we use the Local Search technique for better exploration in the small range, otherwise we use the DE/best/1.

$$R\_limit = \sqrt{\left(\vec{X}_{upper} - \vec{X}_{lower}\right)^T . \left(\vec{X}_{upper} - \vec{X}_{lower}\right)} . 10^{-2}. \tag{4}$$

A brief description of the algorithm is as follows:

4.1 Initialization

We initialize the population randomly in the search space that try to cover the whole search region uniformly as much as possible. Using K-means algorithm we divide the population into clusters with cluster number *cluster_max.*

4.2 Information Exchange among Clusters

For each *g_r* each particle is updated using DE/best/1 and they can exchange information among the particles that belong to that cluster only. If the radius of the cluster becomes less than *R_limit* then we use the local search algorithm in search of better solution within that region. If any cluster has radius less than *R_limit* and does not improve after applying the Local Search technique then the cluster is deleted unless the cluster contains the global best particle. We calculate the percentage change of the function value after each *g_r* and accordingly cluster number is varied. If percentage change is less than *p_ref* then

$$cluster\_no = \max(cluster\_min, prev\_cluster\_no - 1).$$

Otherwise $cluster\_no = \min(cluster\_max, prev\_cluster\_no + 1)$.

4.3 Local Search Technique

We generate some exploiting particles (*exploiters*) to exploit information from the suspected optimum position. When a cluster radius is less than *R_limit* then almost all the particles in that cluster are very close to each other, so if we do Local Search around each particle then it will be some sort of wastage of FEs. So we take only best 40% particles of that cluster and new particles are randomly generated around these 40% of old particles within a radius *r_exploit*. The *r_exploit* is small for the best fit particle and comparatively large for the worst particle among those 40% of particles. The number of *exploiters* generated is varied according to the fitness of the particles. We used

$$exploiters_i = round\begin{pmatrix} (exploiters\_max - exploiters\_min)\left(1 - \left(\dfrac{f_i - best}{worst - best}\right)^2\right) \\ + exploiters\_min \end{pmatrix}, \quad (5)$$

Where f_i is the fitness value of that particle and *best* and *worst* are the fitness value of the best fit and worst fit particle respectively. In case of *r_exploit* we used

$$r\_exploit = (1 - k^d).\frac{(f_i - best)}{worst - best} + r\_ini, \text{ Where } k = \frac{best}{worst}. \quad (6)$$

5 Parameters

$NP = 150$, $cluster\_min = 2$, $cluster\_max = 10$, $exploitrs\_max = 7$, $exploiters\_min = 2$, $d = 1$, $g\_r = 10$, $F = 0.5(rand(0,1) + 1)$, $Cr = 0.9$, $r\_ini = 5e\text{-}03$, $p\_ref = 5$.

6 Experimental Results

We have applied our algorithm on the 25 benchmark problem of CEC-2005 and the dimensionalities of those functions are 10 and 30. We have run the algorithm 25 times on each function and calculated the mean and standard deviation (std) from that results.

Table 1. Error values for dimension $n = 10$

Algorithm	Func1	Func2	Func3	Func4	Func5	Func6	Func7
	Mean Std	Mean Std	Mean Std	Mean Std	Mean Std	Mean Std	Mean Std
GCMA	5.14e-09 1.82e-09	5.31e-09 1.77e-09	4.94e-09 1.45e-09	1.79e+06 4.66e+06	6.57e-09 1.88e-09	5.41e-09 1.81e-09	**4.91e-09** **1.68e-09**
SADE	**0.00e+00** **0.00e+00**	1.04e-13 5.11e-13	1.67e-05 3.12e-05	1.41e-05 7.09e-05	1.2e-02 1.4e-02	1.20e-08 1.93e-08	1.99e-02 1.07e-02
IDE-SR	**0.00e+00** **0.00e+00**	**0.00e+00** **0.00e+00**	**0.00e+00** **0.00e+00**	**0.00e+00** **0.00e+00**	**3.63e-12** **1.99e-12**	**0.00e+00** **0.00e+00**	4.70e-02 2.25e-04

Algorithm	Func8	Func9	Func10	Func11	Func12	Func13	Func14
	Mean Std	Mean Std	Mean Std	Mean Std	Mean Std	Mean Std	Mean Std
GCMA	**2.00e+01** **0.00e+00**	4.49e+01 1.36e+01	4.08e+01 3.35e+01	3.65e+00 1.66e+00	2.09e+02 4.69e+02	4.94e-01 1.38e-01	4.01e+01 3.14e-01
SADE	2.00e+01 5.39e-08	**0.00e+00** **0.00e+00**	4.96e+00 1.69e+00	4.89e+00 6.62e-01	4.50e-07 8.50e-07	2.20e-01 4.00e-02	2.91e+00 2.06e-01
IDE-SR	**2.00e+01** **0.00e+00**	3.75e+00 1.98e+00	**3.70e+00** **1.38e-01**	**1.17e+00** **3.99e-01**	**1.02e-12** **6.35e-11**	**2.10e-01** **1.90e-03**	**2.46e+00** **1.50e-01**

Algorithm	Func15	Func16	Func17	Func18	Func19	Func20	Func21
	Mean Std	Mean Std	Mean Std	Mean Std	Mean Std	Mean Std	Mean Std
GCMA	2.11e+02 1.02e+02	1.05e+02 1.26e+01	5.49e+02 3.49e+02	4.97e+02 2.18e+02	5.16e+02 2.34e+02	4.42e+02 2.03e+02	4.04e+02 1.23e+02
SADE	**3.20e+01** **1.10e+02**	1.01e+02 6.17e+00	1.14e+02 9.97e+00	7.19e+02 2.08e+02	7.05e+02 1.90e+02	7.13e+02 2.01e+02	4.64e+02 1.58e+02
IDE-SR	8.65e+01 1.64e+01	**8.03e+01** **3.56e+01**	**9.17e+01** **2.79e+01**	**3.00e+02** **1.24e-01**	**3.43e+02** **2.47e+01**	**3.17e+02** **1.67e+01**	**3.01e+02** **8.61e+00**

Algorithm	Func22	Func23	Func24	Func25
	Mean Std	Mean Std	Mean Std	Mean Std
GCMA	7.40e+02 5.94e+01	7.91e+02 2.79e+02	8.65e+02 6.39e+02	4.42e+02 3.58e+02
SADE	7.35e+02 9.15e+01	6.64e+02 1.53e+02	**2.00e+02** **0.00e+00**	3.76e+02 3.14e+00
IDE-SR	**7.11e+02** **2.58e+01**	**5.58e+02** **6.46e-01**	2.98e+02 1.84e+01	**3.11e+02** **1.85e+01**

Table 2. Error values for dimension $n = 30$

Algorithm	Func1	Func2	Func3	Func4	Func5	Func6	Func7
	Mean Std	Mean Std	Mean Std	Mean Std	Mean Std	Mean Std	Mean Std
GCMA	5.42e-09 9.80e-10	6.22e-09 8.95e-10	**5.55e-09** **1.09e-09**	1.11e+04 3.02e+04	**8.62e-09** **8.53e-10**	**5.90e-09** **1.61e-09**	**5.31e-09** **1.41e-09**
SADE	**0.00e+00** **0.00e+00**	9.72e-08 4.86e-07	5.05e+04 1.58e+05	5.82e-06 1.45e-05	7.88e+02 1.24e+03	2.12e+01 1.34e+01	8.27e-03 1.14e-02
IDE-SR	**0.00e+00** **0.00e+00**	**1.64e-13** **6.69e-14**	6.38e+02 1.59e+02	**2.36e-02** **5.34e-02**	6.21e+02 4.05e+01	1.17e+00 1.45e+00	5.21e-06 1.57e-05

Algorithm	Func8	Func9	Func10	Func11	Func12	Func13	Func14
	Mean Std	Mean Std	Mean Std	Mean Std	Mean Std	Mean Std	Mean Std
GCMA	2.01e+01 2.79e-01	9.38e-01 1.18e+00	1.65e+00 1.35e+00	5.48e+00 3.13e+00	4.43e+04 2.19e+05	2.49e+00 5.13e-01	1.29e+01 4.19e+01
SADE	2.01e+01 5.73e-03	**2.27e-15** **1.14e-14**	3.58e+01 6.08e+00	2.66e+01 1.13e+00	8.73e+02 9.34e+02	1.21e+00 1.34e-01	1.24e+01 2.59e-01
IDE-SR	**2.01e+01** **1.35e-05**	1.41e-02 1.84e-02	**1.12e+00** **5.21e-02**	**4.99e+00** **1.25e+00**	**1.32e+02** **9.71e+01**	**1.04e+00** **7.39e-01**	**1.14e+01** **3.44e-01**

Algorithm	Func15	Func16	Func17	Func18	Func19	Func20	Func21
	Mean Std	Mean Std	Mean Std	Mean Std	Mean Std	Mean Std	Mean Std
GCMA	2.08e+02 2.75e+01	**3.50e+01** **2.04e+01**	2.91e+02 1.93e+02	9.04e+02 2.88e-01	9.04e+02 2.71e-01	9.04e+02 2.48e-01	5.00e+02 1.31e-13
SADE	3.28e+02 9.65e+01	1.38e+02 1.70e+01	1.51e+03 9.63e+02	9.54e+02 3.44e+01	8.46e+02 6.22e+02	2.02e+03 8.77e+02	1.73e+03 5.12e+02
IDE-SR	**1.15e+02** **2.69e+00**	7.51e+01 1.53e+01	**9.42e+01** **1.17e+01**	**8.49e+02** **3.86e+00**	**8.46e+02** **3.53e+00**	**8.30e+02** **3.92e+00**	**5.00e+02** **3.26e-15**

Algorithm	Func22	Func23	Func24	Func25
	Mean Std	Mean Std	Mean Std	Mean Std
GCMA	8.03e+02 1.86e+01	5.34e+02 2.22e-04	9.10e+02 1.48e+02	2.11e+02 9.21e-01
SADE	1.58e+03 4.52e+02	5.21e+02 2.49e+01	2.00e+02 1.25e-02	5.00e+02 5.68e-03
IDE-SR	**5.45e+02** **4.43e+00**	**4.98e+02** **1.72e+01**	**2.00e+02** **5.61e-03**	**2.07e+02** **1.15e+00**

7 Conclusion

In this paper, we have proposed an Informative Differential Evolution with Self Adaptive Re-clustering Technique for solving global optimization problems. The cluster number is changed in a self adaptive way. The proposed Local Search technique also takes the advantage of self-adaptive behaviour of the algorithm to change the *R_limit* parameter which is crucial for the performance of the *Local Search* algorithm and thus for the performance of IDE-SR.

References

1. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In: Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ, pp. 1942–1948 (1995)
2. Dorigo, M., Stützle, T.: Ant Colony Optimization. MIT Press, Cambridge (2004)
3. Holland, J.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor (1975)
4. Farmer, J.D., Packard, N., Perelson, A.: The Immune System, Adaptation and Machine Learning. Physica D 22, 187–204 (1986)
5. Karaboga, D.: An Idea Based on Honey Bee Swarm for Numerical Optimization, technical REPORT-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department (2005)
6. Passino, K.M.: Biomimicry of Bacterial Foraging for Distributed Optimization and Control. IEEE Control Systems Magazine 22, 52–67 (2002)
7. Storn, R., Price, K.V.: Differential Evolution-A simple and efficient Heuristic for Global Optimization over continuous Spaces. Journal of Global Optimization 11, 341–359 (1997)
8. Ilonen, J., Kamarainen, J.K., Lampinen, J.: Differential Evolution Training Algorithm for Feed- Forward Neural Networks. Neural Processing Letters 7, 93–105 (2003)
9. Storn, R.: Differential evolution design of an IIR-filter. In: Proceedings of IEEE Int. Conference on Evolutionary Computation, ICEC 1996, pp. 268–273. IEEE Press, New York (1996)
10. Suganthan, P.N., Hansen, N., Liang, J.J., Deb, K., Chen, Y.-P., Auger, A., Tiwari, S.: Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization. Nanyang Technological University, Tech. Rep. (2005)
11. Qin, A.K., Huang, V.L., Suganthan, P.N.: Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Trans. on Evolutionary Computations, 398–417 (April 2009), doi:10.1109/TEVC.2008.927706
12. Auger, A., Hansen, N.: A Restart CMA Evolution Strategy With Increasing Population Size. In: Proceedings of the IEEE Congress on Evolutionary Computation, Piscataway, NJ, USA, vol. 2, pp. 1769–1776. IEEE Press (2005)
13. Das, S., Suganthan, P.N.: Differential Evolution: A Survey of the State-of-the-art. IEEE Trans. on Evolutionary Computation 15(1), 4–31 (2011)
14. Mallipeddi, R., Suganthan, P.N., Pan, Q.K., Tasgetiren, M.F.: Differential evolution algorithm with ensemble of parameters and mutation strategies. Applied Soft Computing 11(2), 1679–1696 (2011)
15. Ghosh, A., Das, S., Chowdhury, A., Giri, R.: An improved differential evolution algorithm with fitness-based adaptation of the control parameters. Inf. Sci. 181(18), 3749–3765 (2011)
16. Das, S., Abraham, A., Chakraborty, U.K., Konar, A.: Differential Evolution Using a Neighborhood-Based Mutation Operator. IEEE Transactions on Evolutionary Computation 13(3), 526–553 (2009)

Differential Evolution for Optimizing the Hybrid Filter Combination in Image Edge Enhancement

Tirimula Rao Benala[1], Satchidananda Dehuri[2],
G.S. Surya Vamsi Sirisetti[1], and Aditya Pagadala[1]

[1] Anil Neerukonda Institute of Technology and Sciences
Sangivalasa, Visakhapatnam, Andhra Pradesh, India
b.tirimula@gmail.com, suryavamsy@ymail.com,
aditya_pagadala@yahoo.com
[2] Department of Information & Communication Technology
Fakir Mohan University, Vyasa Vihar, Balasore-756019, India
satchi.lapa@gmail.com

Abstract. Image edge enhancement is the art of enhancing the edge of significant objects in an image. The proposed work uses the concept of hybrid filters for edge enhancement whose optimal sequence is to be found by differential evolution. Its unbiased stochastic sampling and bench-mark results in a quite many number of applications ignited us to use for the aforesaid purpose and motivated for further research. The major five mutational variants of differential evolution employing the binomial crossover have been used in the proposed work which and have been tested over both standard images and medical images. Our extensive experimental studies produce encouraging results.

Keywords: Image edge enhancement, hybridized smoothening filter, differential evolution.

1 Introduction

In the modern era, digital images have been widely used in an aggrandizing number of applications and the effort on edge enhancement has been focused mostly to improve visual perception of images that are unclear (blurred). Edges are the representations of the discontinuities of image intensity functions. For processing these discontinuities in an image, an efficient edge enhancement technique is essential [5]. These edge enhancement techniques fall under two categories: smoothening filters and sharpening filters [2].

Smoothening filters are used for blurring and noise reduction [6]. Noise reduction can be accomplished by blurring with linear filters (mean, median and mode) and non-linear filters (circular, pyramidal and cone) [8]. Sharpening filters (Laplacian, Sobel, Prewitt and Robert filters) are used to highlight fine details in an image or to enhance details that have been blurred but because of their results of complexity and image quality, smoothening filters are used which involves simple subtractive smoothened image concept which reduces complexity and makes the images look sharper than they really are. This can also be done with the help of a new filter called

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 35–42, 2011.
© Springer-Verlag Berlin Heidelberg 2011

the hybridized smoothening filter (a sequence of smoothening filter) [8]. The optimal magnitude (for different combinations of smoothening filters) of the hybrid filter is found by using the differential evolution (DE) and is compared to that obtained by the different variants of it. Hybrid filters, its optimization using DE is explained in Section 2. Section 3 discusses our proposed framework. In Section 4, experiments have been conducted and their corresponding results obtained are discussed. Conclusions and further study is given in Section 5.

2 Hybrid Filters and Differential Evolution Algorithm

2.1 Hybrid Filters

Hybrid filter is defined as the series of existing filters (Smoothening filters) to optimize the magnitude of the image [8]. It can efficiently remove large amounts of mixed Gaussian and impulsive noise besides preserving the image details. In this approach, hybrid filter is taken as a combination of smoothening filters (for e.g. 1-2-3-4-5-6 i.e. suppose 1-mean, 2-median, 3-mode, 4-circular, 5-pyramidal, 6-cone. The output of mean filter is taken as input for median filter and the output of median filter is given as input to the next and so on). This hybrid filter yields optimal threshold values using clustering algorithm.

2.2 Differential Evolution

DE is an evolutionary optimization technique, which is simple, significantly faster and robust at numerical optimization and likely chances of finding true global optimum. The stages involved are:

(i) **Initialization.** The DE starts with the creation of an initial population, normally at random.

(ii) **Mutation.** New individuals are created applying the mutation operations. At each generation, for each individual of the population a new solution *(*v_i^{t+1}*)* is created using a weighted vector difference between two other individuals, selected randomly from the current population. The available schemes use the notation DE/α/γ/β where α represents the way in which individuals are selected from the current population. They can be selected either randomly *(rand)* or as the best individual from the current population *(best)*. The γ is the value of difference vector pairs used, which normally is 1 or 2. *t* represents the tth generation and *r1, r2,* etc. are randomly selected individuals from the tth generation population. F value is the scaling factor, which controls the differential variation (mutation). The different types of mutations are listed in table1.

$$v_i^{t+1} = x_{r3}^t + F(x_{r1}^t - x_{r2}^t) \quad (1)$$

(iii) **Crossover.** The crossover which has been applied here is of binomial type which follows the scheme given in equation 2. For each individual x_i^t, a trial individual u_i^{t+1} is generated by using equation 2 which involves using a crossover constant *cr* [10]

$$u_i^{t+1} = \begin{cases} v_i^{t+1} & \text{if (rand < cr) || j==rand(0,D)} \\ x_i^t & \text{else if(rand> cr)} \end{cases} \tag{2}$$

Table 1. Mutation Schemes [4]

Scheme name	Mutation Definition
DE/ran/1/β	$v_i^{t+1} = x_{r3}^t + F(x_{r1}^t - x_{r2}^t)$
DE/best/1/ β	$v_i^{t+1} = x_{best}^t + F(x_{r1}^t - x_{r2}^t)$
DE/ran/2/ β	$V_i^{t+1} = x_{r5}^t + F(x_{r1}^t - x_{r2}^t + x_{r3}^t - x_{r4}^t)$
DE/best/2/ β	$v_i^{t+1} = x_{best}^t + F(x_{r1}^t - x_{r2}^t + x_{r3}^t - x_{r4}^t)$
DE/rantobest/1/ β	$v_i^{t+1} = x_{r3}^t + F(x_{best}^t - x_{r3}^t) + F(x_{r1}^t - x_{r2}^t)$

(iv) Selection. The child individual x (i, t+1) is selected between each pair of x (i, t) and u (i, t+1) by using greedy selection criterion:

$$x_i^{t+1} = \begin{cases} u_i^{t+1} & \text{if } F(u_i^{t+1}) < F(x_i^t) \\ x_i^t & \text{other wise,} \end{cases} \tag{3}$$

where, F is the fitness function.

3 Framework

3.1 Initialization of Population

First optimal magnitudes of the image are obtained by using adaptive thresh-holding with foreground and background clustering algorithm [9] as described below:

One background and one foreground are assumed, K=2, i.e. only two clusters are considered, which makes the overall implementation easy.

- **(i) Region selection.** Divide the document into all-inclusive mutually exclusive sub-regions. Select the document sub-region for which the threshold be computed, and a region containing the sub-region that will be used to determine the threshold sub-region. For example, the region may consist of N contiguous scan lines, where the sub-region is the center M scan lines, with M<N.
- **(ii) Initialization.** Initialize the background cluster mean and the foreground cluster mean to be the same as computed for the previous sub-region. If there is not one, set the two initial cluster means with a large separation between them. For each pixel inside the region, iterate between steps (iii) and (iv).
- **(iii) Pixel Assignment**. Assign each pixel to the nearest cluster.
- **(iv) Cluster Mean Update.** After each new pixel assignment update the relevant cluster mean. If they are over the range, set them to maximum range.
- **(v) Threshold Calculation.** After all pixels in the region have been assigned, set the threshold for the sub-region equal to the average of the foreground and background cluster means. After obtaining the optimal magnitude of the image we take the random population with the magnitudes of different combinations, which are closer to the obtained optimal magnitude.

3.2 Using Differential Evolution for Finding Optimal Hybrid Filter Combination

(i) Population Generation. Here, we randomly generate the required number of population. Each a population member is nothing but the hybrid filter combination that we intend to optimize.

(ii) Mutation Operation. Using any of the mutation schemes given in table 1, the mutation vector v is found out for every member of population. Hence the mutation vector v, now has new hybrid filter combinations.

(iii) Crossover Operation. Using the binomial Crossover scheme given in equation 2, the crossover vector u is generated for every member of the population. The Crossover constant *cr* has been taken as 0.5.

(iv) Selection. The optimal combinations for next generations are selected in this stage. This selection involves selecting in between the crossover individual and the individual of the previous generation. This is done by using the fitness evaluation as described in step (v). An individual 'n' is selected based on equation 7 in which optimal magnitude of image called as rand (x_n) is subtracted from its fitness value. In this application, F (x_n) has to be as small as possible since that hybrid filter combination would yield a magnitude closer to the optimal magnitude.

(v) Calculate the Fitness Function: An individual's fitness is measured by the sum of intensities of edges E (k) in an enhanced image, because a gray image with a visual good contrast includes many intensive edges.

$$\delta h_k(x, y) = g_k(x+1, y-1) + g_k(x+1, y) + g_k(x+1, y+1) - g_k(x-1, y-1) - g_k(x-1, y) - g_k(x-1, y+1) \tag{4}$$

$$\delta v_k(x, y) = g_k(x-1, y+1) + g_k(x, y+1) + g_k(x+1, y+1) - g_k(x-1, y-1) - g_k(x, y-1) - g_k(x+1, y-1) \tag{5}$$

$$E(k) = \Sigma_x \Sigma_y \sqrt{(\delta h(x, y)^2 + \delta v(x, y)^2)} \tag{6}$$

$$F(x_n) = E(k) - \text{rand}(x_n) \tag{7}$$

4 Experimentations and Results

The images camera man [3], Lena [3], peppers [3], MRI-Brain and MRI-Spine (courtesy: Vijaya Diagnostics, Visakhapatnam) along with their corrupted noisy versions were taken as references for our work. The noisy versions include noises of Gaussian noise 0.0mean-0.1standard deviation, 0.1mean-0.1standard deviation, 0.05mean-0.05 standard deviation and Salt and Pepper noise 0.02standard deviation, 0.04standard deviation and0.06 standard deviation. The following table gives the details of these images. The entire work has been carried out in java using NetBeans IDE 6.8 as platform. Matlab has been used to add noise to the original images. Further details of the images have been listed under Table 2.

Table 2. Details about the Images

Image	Format	Size (pixels)	Resolution (dpi)	Bit Depth
Camera man	JPG	512 x 512	96 x 96	8
Lena	PNG	512 x 512	72 x 72	32
Peppers	JPG	512 x 512	72 x 72	24
MRI-Brain	JPG	512 x 512	96 x 96	24
MRI-Spine	JPG	512 x 512	96 x 96	24

The order in which each of the images and its noisy versions has been given below is as follows:

1. Gaussian 0.0m-0.1d
2. Gaussian 0.1m-0.1d
3. Gaussian 0.05m-0.05d
4. Original Image
5. Salt and Pepper 0.02d
6. Salt and Pepper 0.04d
7. Salt and Pepper 0.06d

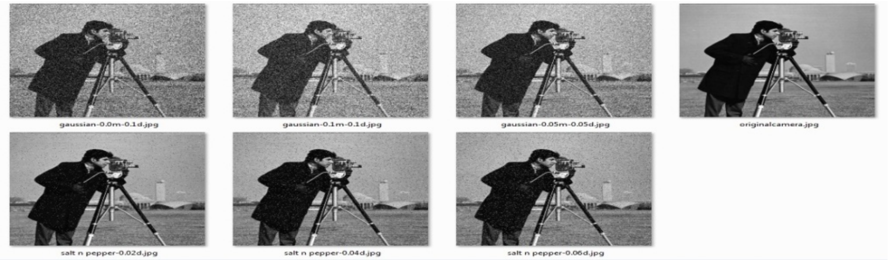


Fig. 1. Camera Man

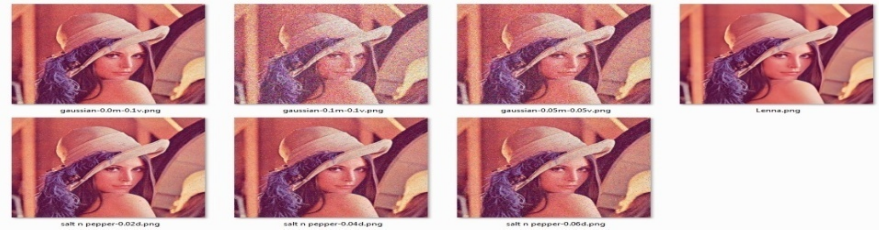


Fig. 2. Lina

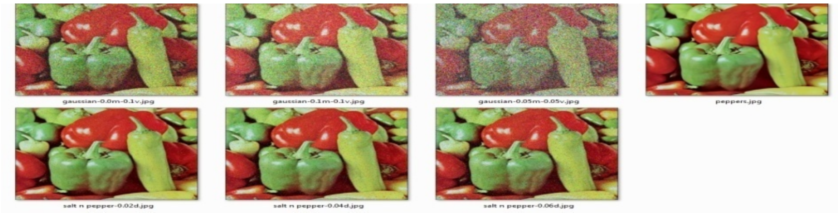

gaussian-0.0m-0.1v.jpg gaussian-0.1m-0.1v.jpg gaussian-0.05m-0.05v.jpg peppers.jpg
salt n pepper-0.02d.jpg salt n pepper-0.04d.jpg salt n pepper-0.06d.jpg

Fig. 3. Peppers

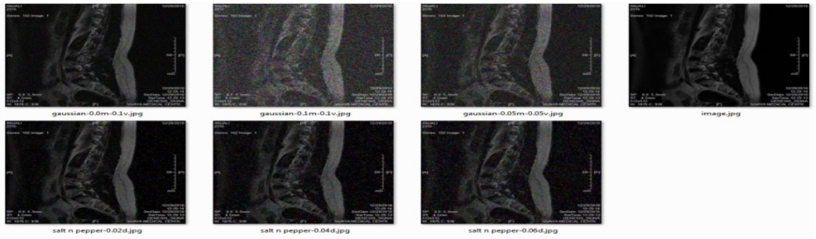

gaussian-0.0m-0.1v.jpg gaussian-0.1m-0.1v.jpg gaussian-0.05m-0.05v.jpg image.jpg
salt n pepper-0.02d.jpg salt n pepper-0.04d.jpg salt n pepper-0.06d.jpg

Fig. 4. MRI- Brain

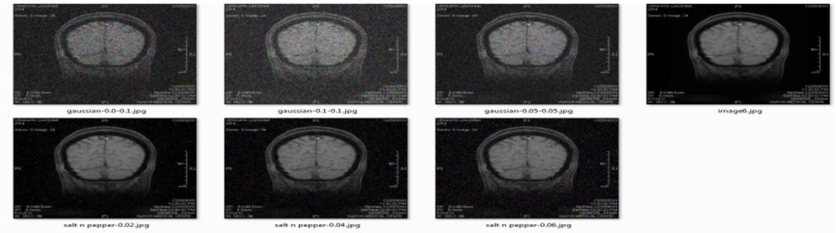

gaussian-0.0-0.1.jpg gaussian-0.1-0.1.jpg gaussian-0.05-0.05.jpg image6.jpg
salt n pepper-0.02.jpg salt n pepper-0.04.jpg salt n pepper-0.06.jpg

Fig. 5. MRI- Spine

4.1 Observation Tables

For observing the efficiency of the various algorithms used we have adopted various factors like:

1. Standard Deviation (d)
2. Mean Square Error (ms)
3. Mean Absolute Error (ma)

Tables 3 gives a remarkable view of all filters applied on MRI-Brain image. Relying on the values in the table one can ascertain that results obtained by hybrid filters are closer to original magnitudes of the images. It can be inferred from the table that the results obtained by the different algorithms vary from image to image. Any one of the algorithms might give better results for a given image but, it doesn't hold for all kinds of images. The performance of these algorithms varies with the kind of images that they are applied to like for example the noise that it has. These algorithms always produce optimal results than other traditional approaches.

Table 3. MRI- Brain

Images			DE/best/1/ bin	DE/best/2 /bin	DE/rand/1/ bin	DE/rand/2/ bin	DE/rantobest/ 1/bin
Salt and Pepper Noise	0.06 d	ma	30.8	22.1	24.2	24.5	21.6
		ms	130.06	70.93	105.8	91.35	73.32
		d	36.06	26.63	32.526	30.224	27.07
	0.04 d	ma	8.4	3.5	2.9	5.0	6.7
		ms	13.04	1.95	1.289	3.44	5.93
		d	11.41	4.41588	3.5916	5.8651	7.700
	0.02 d	ma	12.1	10.9	6.6	14.2	9.6
		ms	26.58	23.57	5.34	34.38	20.74
		d	16.30	15.35	7.3075	18.541	14.401
Gaussian Noise	0.05 m-0.05 d	ma	20.1	9.2	14.4	15.5	23.6
		ms	76.63	12.0	41.4999	38.9299	107.16
		d	27.68	10.954	20.371	19.730	32.735
	0.1 m-0.1 d	ma	33.6	24.5	54.2	31.4	33.2
		ms	202.299	110.78	419.64	189.32	178.97
		d	44.97	33.285	64.7796	43.51	42.30602
	0.0 m-0.1 d	ma	16.9	23.4	14.7	27.8	23.4
		ms	73.83	88.979	45.69	112.239	80.76
		d	27.17	29.829	21.375	33.502	28.418
Original	ma		23.3	27.2	22.8	24.9	24.9
	Ms		75.4899	87.16	111.58	102.15	75.85
	D		27.47	29.522	33.403	31.960	27.540

5 Conclusion and Future Work

As processing time is not an issue and only efficiency is the prime criteria in the field of edge enhancement, we have been exploring the scope of DE in this field. We have done our research in the direction of edge enhancement by hybrid filters through DE. Different variants of DE have been found to provide better image quality compared to smoothening filters and baroque results in displaying the output image with or without noise. We extend connotations to our work with Artificial Immune System (AIS), Bacterial foraging optimization algorithm, Neuro Fuzzy, Neuro Genetic, Simulated Annealing and fuzzy logic.

References

1. Benela, T.R., Jampala, S.D., Villa, S.H., Konathala, B.: A novel approach to image edge enhancement using artificial bee colony algorithm for hybridized smoothening filters. In: IEEE Conference on BICA, India (2009) ISBN 978-1-4244-5612-3/09
2. Gonzalez, W.: Digital Image Processing, 2nd edn. Prentice Hall, Englewood Cliffs (2001)
3. Image database, `http://www.imageprocessingplace.com/root_files_V3/image_databases.html`
4. Liu, Z., Zhang, Y., Ning, Z., Zhang, Y., Guan, Y.: Differential Evolution based on Improved Mutation Strategy. In: 2nd International Conference on Computer Engineering and Technology. IEEE (2010) 978-1-4244-6349-7/10
5. Maximiano, M.S., Vega-Rodríguez, M.A., Gómez-Pulido, J.A., Sánchez-Pérez, J.M.: A Hybrid Differential Evolution Algorithm to Solve a Real-World Frequency Assignment Problem. In: Proceedings of the International Multi Conference on ISBN Computer Science and Information Technology, pp. 201–205 (2008) ISSN 1896-7094, ISBN 978-83-60810-14-9
6. Ojima, Y., Kirigaya, S., Wakahara, T.: Determining Optimal Filters for Binarization of Degraded Grayscale Characters Using Genetic Algorithms. In: Proceedings of Eighth International Conference on IEEE Document Analysis and Recognition, vol. 2, pp. 555–559 (2005) ISBN 0-7695-2420-6
7. Price, K., Storn, R.: Differential Evolution: A Simple Evolution Strategy for Fast Optimization. Dr. Dobb's Journal 22(4), 18–24 (1997)
8. Rao, B.T., Dehuri, S., Dileep, M., Vindhya, A.: Swarm Intelligence for Optimizing Hybridized Smoothing Filter in Image Edge Enhancement. In: Panigrahi, B.K., Das, S., Suganthan, P.N., Dash, S.S. (eds.) SEMCCO 2010. LNCS, vol. 6466, pp. 370–379. Springer, Heidelberg (2010)
9. Savakis, A.E.: Adaptive Document Image Thresh-holding Using foreground and Background clustering. In: IEEE Proceedings of International Conference on Image Processing ICIP, vol. 3, pp. 785–789 (1998) ISBN 0-8186-8821-1
10. Zaharie, D.: A Comparative Analysis of Crossover Variants in Differential Evolution. In: Proceedings of the International Multi Conference on Computer Science and Information Technology, pp. 171–181 (2007) ISSN 1896-7094
11. Zhu, G., Chen, Z.: A Differential Evolution Optimization Approach to Solve the Pick-and-Placing Problem. In: Fifth International Conference on Natural Computation (2009) 978-0-7695-3736-8/09, doi:10.1109/ICNC.2009.153

Scheduling Flexible Assembly Lines Using Differential Evolution

Lui Wen Han Vincent and S.G. Ponnambalam

Monash University, Sunway Campus, 46150 Petaling Jaya, Malaysia
whlui1@student.monash.edu
sgponnambalam@eng.monash.edu.my

Abstract. This paper investigates the performance of Differential Evolution (DE) in solving a Flexible Assembly Line (FAL) scheduling problem. Using a mathematical model developed in literature, the DE algorithm is implemented with the objectives of minimizing the sum of Earliness/Tardiness (E/T) penalties and maximizing the balance of the FAL. Experimental results have shown that DE is capable of solving the FAL scheduling problem effectively. Furthermore, a comparison with similar work in literature which employs Genetic Algorithm (GA) shows that DE produces a better solution.

1 Introduction

A Flexible Assembly Line (FAL) is a unidirectional production flow line which comprises a set of assembly stations inter-linked with an automated material handling system. One of the main problems in the effective operation of FALs is scheduling. Effective scheduling in FALs is important is important to optimize the allocation of resources so that production costs can be lowered. FAL scheduling is a formidable problem because it allows the assignment of one operation to multiple machines and multiple operations on one machine. In practice, scheduling remains a difficult problem as there is a need to address both routing and operation flexibility. It has been proven that even a simple two-group scheduling problem is NP-hard [1].

Various techniques have been proposed to solve the scheduling problem. These techniques can be generally classified as efficient optimal methods, enumerative optimal methods, and heuristic methods [6]. While the first method produces an optimal solution, it is often limited to a small class of problems and entails a substantial computational effort. Enumerative methods, such as branch-and-bound enumerates a set of all possible solutions. On the other hand, heuristic methods such as Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Simulated Annealing (SA), and Ant Colony Optimization (ACO) attempt to find a near optimal solution instead of the best solution. This method has captured the interests of researchers because they have produced promising results while remaining computationally efficient and easy to implement.

For a FAL, researchers are often interested to generate a schedule which maximizes the balance of the FAL. In other words, researchers want to determine the

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 43–50, 2011.
© Springer-Verlag Berlin Heidelberg 2011

operation assignment to workstations which optimizes the performance of the FAL. In recent years, minimizing the Earliness/Tardiness (E/T) penalties has become equally important due to the awareness of the Just-in-Time (JIT) production concept. Guo, et al. [4] proposed a GA-based-optimization model to solve a FAL scheduling problem with the objectives of minimizing the sum of E/T penalties and maximizing the balance of the FAL. Guo, et al. [5] then further expanded the problem by adding constraints of workstation revisiting and work sharing. However, the performance of these models could not be compared with others as similar research has not been published elsewhere. Moreover, the effects of various uncertainties in the FAL, such as unpredictable customer behavior and machine breakdown were not considered.

In this paper, we are interested in solving the FAL scheduling problem based on the work in [4] using Differential Evolution (DE), a recent heuristic search technique. We would also like to provide a comparison between DE and GA in solving the same scheduling problem.

1.1 Differential Evolution

DE is a parallel direction search technique which explores the search space in a stochastic manner through a population of D-dimensional parameter vectors [7] $x_{i,\ k}$, where $i = 1, 2,\ldots, N$ and k is the generation number of the algorithm. In each generation, the entire population is targeted for replacement by applying mutation, crossover and acceptance operators. In this paper, focus will be placed on how DE is applied to solve the FAL scheduling problem. Comprehensive surveys on DE [2] are available in literature.

2 Problem Formulation

The notations adopted in this paper are summarized in Table 1. Consider a two order FAL scheduling problem, with the orders denoted as i and j. For such a problem, we consider five arrangements or modes in which the orders can be schedules, as shown in Fig. 1. If the value of t_1 is such that order i is completed, mode (c) becomes mode (e). Hence, mode (e) is a subset of mode (c). Similarly, mode (d) is a subset of modes (a) and (b).

2.1 Objective Function

The quality of the generated schedule is measured through two criteria, namely the weighted sum of E/T penalties and the balance of the FAL. The former criterion is defined as

$$\min Z\left(S_{nl}, X_{nlkj}\right) = \sum_{n=1}^{p} \left(\alpha_n \cdot (F_n - D_n) \cdot \lambda_n + \beta_n \cdot (D_n - F_n)(1 - \lambda_n)\right) . \qquad (1)$$

Table 1. Nomenclature

Notation	Description
P_n	The n^{th} production order
O_{nl}	The l^{th} operation of the n^{th} order
M_{kj}	The j^{th} machine of the k^{th} machine type
S_{nl}	Actual starting time of operation O_{nl}
X_{nlkj}	Indicates whether operation O_{nl} is assigned to machine M_{kj}. (If so, $X_{nlkj} = 1$. Otherwise, $X_{nlkj} = 0$.)
α_n	Tardiness weight (penalty cost per time unit of delay) of order P_n
β_n	Earliness weight (storage cost per time unit) of order P_n
D_n	Due date of order P_n pre-determined by the customer
F_n	Actual completion time of order P_n
λ_n	Indicates if the tardiness of order P_n is greater than 1. (If so, $\lambda_n = 1$. Otherwise, $\lambda_n = 0$.)
SPT_r	Processing time of the r^{th} scheduling status
SB_r	Balance index of the r^{rh} scheduling status
SS_r	The set of orders involved in the r^{th} scheduling status
n $(n \geq 1)$	The number of orders in the r^{th} scheduling status
PB_n	The balance index of order P_n
o_n	The number of operations of order P_n
OPT_{nl}	The average processing time of operation O_{nl}
ST_{nl}	The standard processing time of operation O_{nl}
EM_{nlkj}	Operator's efficiency to process operation O_{nl} on machine M_{kj}.
η_{nlkj}	Task proportion of operation O_{nl} being assigned to machine M_{kj}. In this paper, task proportion is assumed to be divided equally. For example, if there are nl operations to be processed by machine M_{kj}, $X_{nlkj} = 1$.

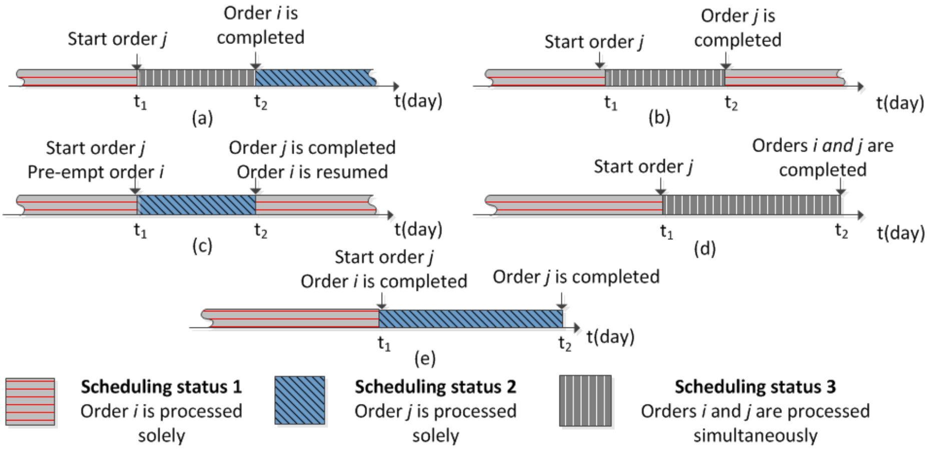


Fig. 1. Scheduling modes of a two-order problem

On the other hand, the latter criterion is defined as

$$\max B\left(S_{n1}, X_{nlkj}\right) = \max \frac{\sum_r SPT_r \cdot SB_r}{\sum_r SPT_r}. \tag{2}$$

The terms SB_r and PB_r are defined as

$$SB_r = \sum_{n, P_n \in SS_r} \frac{PB_n}{n}. \tag{3}$$

$$PB_r = \frac{\sum_i OPT_{nl}}{o_n \cdot \max\left(OPT_{nl}\right)} \tag{4}$$

The two terms above require an additional term OPT_{nl}, which is defined as

$$OPT_{nl} = \frac{ST_{nl}}{\sum_{kj} X_{nlkj} EM_{nlkj} \eta_{nlkj}}. \tag{5}$$

Finally, converting the balance index to a minimization function, the objective function can be obtained as follows:

$$OBJ\left(S_{n1}, X_{nlkj}\right) = Z\left(S_{n1}, X_{nlkj}\right) + \gamma \cdot \left(1 - B\left(S_{n1}, X_{nlkj}\right)\right). \tag{6}$$

2.2 Assumptions

1) An operation cannot be interrupted once it has started.
2) There is no material shortage, machine breakdown, and absence of operators.
3) The assembly line is empty at time $t = 0$; there is no Work-in-Process (WIP).
4) Operator efficiencies are constant.

2.3 Constraints

1) *Allocation constraint*: Operation O_{nl} can only be assigned to workstations with the suitable machine type. Each operation must be processed at least once.
2) *Precedence constraint*: An operation starts only if its preceding operation is complete.

3 DE Algorithm for FAL Scheduling Problem

In this problem, there are two objectives–to balance the FAL and to minimize the sum of E/T penalties. The first objective depends on the operation assignments to workstations. The second objective depends on the actual completion time of the orders, F_n. However, the value of F_n depends on the operation assignments. Hence, two levels of optimization are performed using DE. Level one DE (DEA1) optimizes the assignment of operations O_{nl} to workstations M_{kj}. Based on these assignments, level two DE (DEA2) optimizes the starting time S_{n1} of each scheduling status.

3.1 Representation

Each parameter vector in DEA1 represents a set of operation assignments for the three scheduling statuses. Each vector consists of three sub-vectors. A sub-vector represents the operation assignment in one scheduling status. The number of parameters in each sub-vector is the number of workstations in the FAL. Each parameter represents a workstation; its value denotes the operation assigned to that workstation. Since there are two machine types in the FAL, each sub-vector can be further divided into two parts. Each part represents one machine type. Hence, there is a total of six parts in each parameter vector.

For DEA2, each parameter vector represents the starting time of each scheduling status. If the value of t_1 is known, the starting time of each scheduling status can be determined. Hence, t_1 is the only parameter of the vector.

Parameter vectors for scheduling problems are naturally represented as integer vectors. However, this approach cannot be used directly in DE because the mutation operator produces floating-point vectors. Hence, the sub-range encoding scheme [9] is applied to map the floating-point vectors to integer vectors. In this approach, six sub-range arrays are required to represent the set of operations allowed in each part of the vector. Then, the floating-point vectors are mapped to their corresponding operation numbers depending on the sub-range index that they fall into.

3.2 Algorithm

- **Step 1: Generate initial population for DEA1 and DEA2, and select values for the control parameters.**
- **Step 2: Evaluate fitness of DEA2 parameter vectors and determine optimal scheduling mode.**
- The fitness of DEA2 parameters is determined using (1). The mode with the smallest sum of E/T penalties is the optimal scheduling mode.
- **Step 3: Evaluate fitness of DEA1 parameter vectors.**
  Using the optimal scheduling mode, the fitness of parameter vector i in DEA1 is determined by converting (6) to a fitness function so that it can be compared with the results of GA:

$$f = \frac{10}{OBJ(S_{n1}, X_{nlkj}) + 1}. \tag{7}$$

- **Step 4: Repeat Steps 2 and 3 for all parameter vectors in DEA1.**
- **Step 5: Apply the mutation, crossover and acceptance operators for both DEA1 and DEA2.**
  For mutation, using the notation of Das, et al. [2], the DE/best/2/bin scheme is used. Furthermore, a time-varying mutation scale factor F is used [3]:

$$F = (F_{\max} - F_{\min}) \times (GEN - k)\,. \tag{8}$$

$F_{\max}$ and $F_{\min}$ are the upper and lower bounds of the mutation scale factor respectively. *GEN* represents the number of generations.

- **Step 6: Return to Step 2. Repeat the process until the termination criterion is satisfied.**

  The termination criterion in this paper is set as the number of generations. DE stops once the number of generations is equal to *GEN*.

4 Results and Discussion

Two experiments have been conducted. In both experiments, the FAL have eight working hours per day. Each experiment can be further divided into different cases. Experiment 1 comprises two cases while Experiment 2 consists of three. In Experiment 1, both cases have the same order due dates. Case 1 allows only one operation to be assigned to a workstation whereas Case 2 allows up to two operations to be assigned to one workstation. In Experiment 2, up to two operations can be assigned in all cases. However, each case has different order due dates: 28/32 for Case 1, 32/28 for Case 2, and 32/20 for case 3.

The data set for both experiments are shown in Table 2 and 3. Table 4 shows the final obtained values of the control parameters while Table 5 presents the fitness value of (7). The best fitness and average fitness are the best and average values respectively from 10 test runs. In Experiment 1 and Case 1 of Experiment 2, DE achieves a better performance compared to GA in terms of the best and average fitness. In Case 2 of Experiment 2, DE achieves a better performance in terms of the best fitness value.

Fig. 2 shows the performance curve of a test run using DE in terms of its best fitness. This curve shows that DE has a good convergence rate. Except for Case 1 of Experiment 2 and Case 2 of Experiment 1, the final fitness values are obtained in approximately 100 generations.

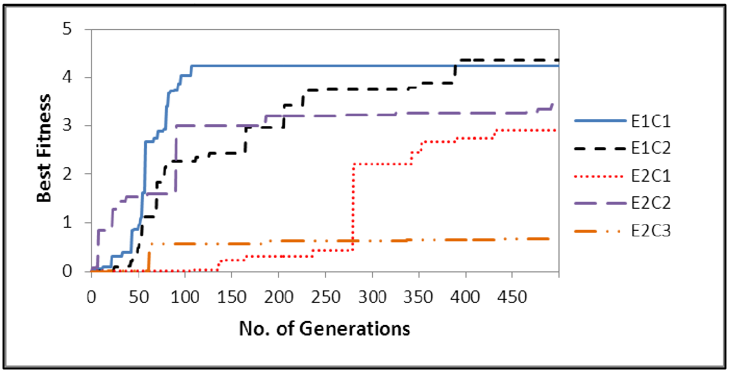


Fig. 2. Performance curve of DE algorithm

Table 2. Data sets for Experiment 1 (left side of /) and 2 (right side of /)

Order	Order size	Due date	Tardiness weight	Earliness weight	Operation	Machine type required	Standard time (s)
1	2,000	15/28	5,000	100	1	T2/T1	308/308
					2	T1/T1	310/310
					3	T1/T1	150/330
					4	T1/T1	160.350
					5	T2/T2	280/240
					6	T2/T2	320/160
					7	T1/T1	270/300
2	2,000	28/32	3,000	100	8	T2/T1	218/300
					9	T2/T2	156/180
					10	T1/T1	280/240
					11	T2/T2	120/120
					12	T1/T1	200/260

Table 3. Operative efficiencies of operators in experiment 1 (row 1 of workstation *n*) and 2 (row 2 of workstation *n*)

Machine type	Wst	Operation no. of order 1							Operation no. of order 2				
		1	2	3	4	5	6	7	8	9	10	11	12
T1 (E1,E2)	1	0	1	0.95	0.90	0	0	1	0	0	1	0	0.95
		0.90	1	0.95	0.90	0	0	1	0.90	0	1	0	0.95
	2	0	0.75	0.70	0.70	0	0	0.75	0	0	0.80	0	0.70
		0.70	0.75	0.70	0.70	0	0	0.75	0.80	0	0.80	0	0.70
	3	0	0.80	0.80	0.85	0	0	0.80	0	0	0.60	0	0.85
		0.80	0.80	0.80	0.85	0	0	0.80	0.80	0	0.75	0	0.85
	4	0	0.70	0.65	0.60	0	0	0.70	0	0	0.75	0	0.65
		0.70	0.70	0.65	0.60	0	0	0.70	0.70	0	0.75	0	0.65
	5	0	0.85	0.75	0.80	0	0	0.85	0	0	0.95	0	0.80
		0.80	0.85	0.75	0.80	0	0	0.85	0.90	0	0.95	0	0.80
	6	0	0.90	0.85	0.90	0	0	0.90	0	0	0.90	0	0.95
		0.90	0.90	0.85	0.90	0	0	0.90	0.90	0	0.90	0	0.95
	7	0	0.95	1	1	0	0	0.95	0	0	0.85	0	1
		0.95	0.95	1	1	0	0	0.95	0	0	0.85	0	1
T2 (E1)	8	0.90	0	0	0	0.95	0.95	0	0.90	0.95	0	0.95	0
		0.80	0.75	0.75	0.75	0	0	0.80	0.85	0	0.75	0	0.80
T2 (E2)	9	0.80	0	0	0	0.80	0.70	0	0.85	0.90	0	0.70	0
		0	0	0	0	1	0.90	0	0	0.95	0	0.90	0
	10	0.60	0	0	0	0.70	0.65	0	0.70	0.70	0	0.80	0
		0	0	0	0	0.70	0.80	0	0	0.85	0	0.75	0
	11	0.75	0	0	0	0.70	0.80	0	0.80	0.85	0	0.70	0
	12	1	0	0	0	1	0.90	0	0.90	0.90	0	0.85	0
	13	0.95	0	0	0	1	0.90	0	1	0.85	0	1	0
	14	0.90	0	0	0	0.85	1	0	0.95	1	0	0.90	0

Table 4. Obtained final values of control parameters

N (DEA1)	N (DEA2)	*GEN*	$F_{\min}$	$F_{\max}$	C_R
200	30	500	0.3	0.8	0.50

Table 5. Comparison of fitness between DE and GA

Experiment	Best Fitness		Average Fitness	
	DE	GA	DE	GA
Experiment 1 Case 1	**4.03**	1.58	**3.40**	1.58
Experiment 1 Case 2	**4.17**	2.68	**3.54**	2.68
Experiment 2 Case 1	**2.33**	1.42	**1.49**	1.42
Experiment 2 Case 2	3.38	**3.73**	1.87	**3.73**
Experiment 2 Case 3	**1.45**	0.97	0.70	**0.97**

5 Conclusion

The DE algorithm can be used to solve a two-order FAL scheduling problem. Comparing between DE and GA in solving the same problem, experimental results show that DE produces a better solution compared to GA.

References

1. Blazewicz, J., Kovalyov, M.Y.: The complexity of two group scheduling problems. Journal of Scheduling 5, 477–485 (2002)
2. Das, S., Suganthan, P.N.: Differential Evolution: A Survey of the State-of-the-Art. IEEE Trans. Evolutionary Computation 15(1), 4–31 (2011)
3. Das, S., Konar, A., Chakraborty, U.K.: Two improved differential evolution schemes for faster global search. In: GECCO 2005, pp. 991–998 (2005)
4. Guo, Z.X., et al.: A genetic-algorithm-based-optimization model for scheduling flexible assembly lines. The International Journal of Advanced Manufacturing Technology 36, 156–168 (2008)
5. Guo, Z.X., et al.: A genetic-algorithm-based-optimization model for solving the flexible assembly line balancing problem with work sharing and workstation revisiting. IEEE Transactions on Systems, Manufacturing and Cybernetics. Part C, Applicationsand Reviews 38, 218–228 (2008)
6. Marccarthy, B.L.: Addressing the gap in scheduling research: a review of optimization and heuristic methods in production scheduling. International Journal of Production Research 36, 59–79 (2008)
7. Storn, R., Price, K.: Differential evolution- A simple and efficient heuristic for globaloptimization over continuous spaces. Journal of Global Optimization 11, 341–349 (1996)

A Differential Evolution Based Approach for Multilevel Image Segmentation Using Minimum Cross Entropy Thresholding

Soham Sarkar, Gyana Ranjan Patra, and Swagatam Das

ETCE Department, Jadavpur University, Kolkata – 700032, India
{sarkar.soham,gyana.patra}@gmail.com, swagatamdas19@yahoo.co.in

Abstract. Image entropy thresholding is one of the most widely used technique for multilevel thresholding. The endeavor of this paper is to focus on obtaining the optimal threshold points. Several meta-heuristics are being applied in literatures over the decade, for improving the accuracy and computational efficiency of Minimum Cross Entropy Thresholding (MCET) method. In this paper, we have incorporated a Differential Evolution (DE) based approach towards image segmentation. Results are also compared with modern state-of-art algorithms like Particle Swarm Optimization (PSO) and Genetic Algorithm (GA). Further Mean Structural Similarity Index Measurement (SSIM) and Universal Image Quality Index (UIQI) are also being used for performance evaluation.

Keywords: Multilevel Image Segmentation, MCET, DE, MSSIM, UIQI.

1 Introduction

Image segmentation is the process to discriminate object from its background in pixel level and has become the most significant component of image analysis. Over the years segmentation is being applied as a basic step for several computer vision applications like feature extraction, identification, image registration etc. Image segmentation done via bi-level thresholding sub divides the image into two homogenous regions, based on texture, histogram, edge etc, it uses one threshold value, whereas, Image segmentation done via multilevel thresholding splits the image into different classes by selecting multiple threshold points [1].

Entropy based approaches for multilevel image segmentation are widely popular at present day, especially Shannon entropy, Renyi entropy [2], Tsallis entropy [3] and cross entropy [4] etc. Minimum Cross Entropy (MCE) [13] based thresholding has drawn considerable attention of researchers as a complex optimization problem. Recently several successful meta-heuristics like Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Stimulated Annealing (SA), and Bacteria Forging Optimization (BFO) are being applied to achieve the optimum threshold values [5], [6], [7].

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 51–58, 2011.
© Springer-Verlag Berlin Heidelberg 2011

In this paper we have applied Differential Evolution (DE), proposed by Storn [8] for minimization of the Cross Entropy (CE) between the original image and its thresholded version. DE has found an effective paradigm for numerous complex optimization problems during last decade [9], [10], [11]. In addition to that, DE is superior to other algorithms in terms of convergence time. The computational time of the DE for solving image threshold is also being investigated in this article.

The rest of the paper is organized as follows. Section 2 introduces the concept of MCET. Section 3 describes the details of method and formulation of the objective function. A brief introduction of DE is given in the Section 4.In Section 5, the experimental results and comparative performances are presented. Finally the conclusion is made in Section 6.

2 Minimum Cross Entropy Thresholding

The cross entropy was developed by Kullback in [12]. Let $F = \{f_1, f_2 f_3, \ldots.., f_N\}$ and $G = \{g_1, g_2 g_3, \ldots.., g_N\}$ be two probability distributions on the same set. The cross entropy between F and G is defined by

$$D(F,G) = \sum_{i=1}^{N} f_i \; log \frac{f_i}{g_i} \tag{1}$$

The thresholds are being selected by minimizing the MCET between the original image and resulted image. For better understanding of multilevel concept first bi-level thresholding is discussed. Let I be the original image and $h(i), i = 1,2, \ldots.., L$ is the histogram of the corresponding image. L is indicating the number of gray levels. Then the resulted image I_t can be expressed as

$$I_t(x,y) = \begin{cases} \mu(1,t) & I(x,y) < t \\ \mu(t, L+1) & I(x,y) \geq t \end{cases} \tag{2}$$

where t = selected threshold which divides the image into two regions (object and background) and$\mu(a,b) = \sum_{i=0}^{b-1} ih(i) / \sum_{i=0}^{b-1} h(i)$

The cross entropy then calculated by

$$D(t) = -\sum_{i=1}^{t-1} ih(i) \log\left(\frac{i}{\mu(1,t)}\right) + \sum_{i=t}^{L} ih(i) \log\left(\frac{i}{\mu(t, L+1)}\right) D(t) \tag{3}$$

The MCET determines the optimal threshold t^* by minimizing the cross entropy,

$$t^* = \arg\min_t D(t) \tag{4}$$

The computational complexity for determining t^* is $O(nL^2)$. However, it could be time-consuming under the multilevel thresholding scenario. For the n-threshold values, it requires $O(nL^{n+1})$.

3 Recursive MCET

The MCET objective function (4) can be rewritten as

$$D(t) = -\sum_{i=1}^{L} ih(i)\log(i) - \sum_{i=1}^{t-1} ih(i)\log(\mu(1,t)) - \sum_{i=t}^{L} ih(i)\log(\mu(t,L+1)) \qquad (5)$$

Since the first term is constant for a given image, the objective function can be redefined as

$$\delta(t) \; = -\sum_{i=1}^{t-1} ih(i)\log\big(\mu(1,t)\big) - \sum_{i=t}^{L} ih(i)\log\big(\mu(t,L+1)\big)$$

$$= -\left(\sum_{i=1}^{t-1} ih(i)\right)\log\left(\frac{\sum_{i=1}^{t-1} ih(i)}{\sum_{i=1}^{t-1} h(i)}\right) - \left(\sum_{i=t}^{L} ih(i)\right)\log\left(\frac{\sum_{i=t}^{L} ih(i)}{\sum_{i=t}^{L} h(i)}\right)$$

$$= -m^1(1,t)\log\left(\frac{m^1(1,t)}{m^0(1,t)}\right) - m^1(t,L+1)\log\left(\frac{m^1(t,L+1)}{m^0(t,L+1)}\right) \qquad (6)$$

where $m^0\,(a,b) = \sum_{i=1}^{b-1} h(i)$ and $m^1\,(a,b) = \sum_{i=1}^{b-1} ih(i)$ are the zero-moment and first-moment on partial range ofthe image histogram.

The recursive programming technique can be easily applied to multilevel thresholding. Assume n is the number of selected thresholds denoted by $t_1, t_2, t_3 \ldots t_n$. For the convenience of the illustration two dummy threshold $t_0 \equiv 0$, $t_{n+1} \equiv L+1$ and $t_0 < t_1 < \cdots < t_n < t_{n+1}$, are being added. The objective function for multi-level thresholding becomes

$$\delta(t_1, t_2, \ldots\ldots, t_n) = \; m^1(t_{i-1}, t_i)\log\left(\frac{m^1(t_{i-1}, t_i)}{m^0(t_{i-1}, t_i)}\right) \qquad (7)$$

It is found that the computational complexity of (7) is $O(nL^n)$, which is less than $O(nL^{n+1})$ïïIn our approach, DE is used for minimizing $\delta(t)$.

4 Differential Evolution (DE)

DE, a population-based global optimization algorithm, was proposed by Storn in 1995. The i^{th} individual (parameter vector) of the population at generation (time) t is a D-dimensional vector containing a set of Doptimization parameters:

$$\overrightarrow{Z_\imath}(t) = [Z_{i,1}(t), Z_{i,2}(t), \ldots\ldots, Z_{i,D}(t)] \qquad (8)$$

In each generation to change the population members $\overrightarrow{Z_\imath}(t)$ (say), a *donor* vector $\overrightarrow{Y_\imath}(t)$is created. It is the method of creating this donor vector that distinguishes the various DE schemes. In one of the earliest variants of DE, now called DE/rand/1 scheme, to create $\vec{Y_i}(t)$ for each i^{th}member, three other parameter vectors (say the r_1, r_2, and r_3-th vectors such that $r_1, r_2, r_3 \in [1, NP]$and $r_1 \neq r_2 \neq r_3$) are chosen at random from the current population. The donor vector $\vec{Y_i}(t)$ is then obtained multiplying a scalar number F with the difference of any two of the three. The process for the j^{th} component of the i^{th} vector may be expressed as,

$$Y_{i,j}(t) = Z_{r1,j}(t) + F.\left(Z_{r2,j}(t) - Z_{r3,j}(t)\right) \tag{9}$$

A 'binomial' crossover operation takes place to increase the potential diversity of the population. The binomial crossover is performed on each of the D variables whenever a randomly picked number between 0 and 1 is within the Cr value. In this case the number of parameters inherited from the mutant has a (nearly) binomial distribution. Thus for each target vector $\vec{Z_i}(t)$, a trial vector $\vec{R_i}(t)$is created in the following fashion:

$$R_{i,j}(t) = \begin{cases} Y_{i,j}(t) & if\ rand_j(0,1) \leq Cr\ or\ j = rn(i) \\ Z_{i,j}(t) & if\ rand_j(0,1) > Cr\ or\ j \neq rn(i) \end{cases} \tag{10}$$

for $j = 1, 2, \ldots, D$ and $rand_j(0,1) \in [0,1]$is the j^{th} evaluation of a uniform random number generator. $rn(i) \in [1, 2, \ldots \ldots, D]$ is a randomly chosen index to ensures that$\vec{R_i}(t)$ gets at least one component from $\vec{Z_i}(t)$. Finally 'selection' is performed in order to determine which one between the target vector and trial vector will survive in the next generation i.e. at time $t = t + 1$. If the trial vector yields a better value of the fitness function, it replaces its target vector in the next generation; otherwise the parent is retained in the population:

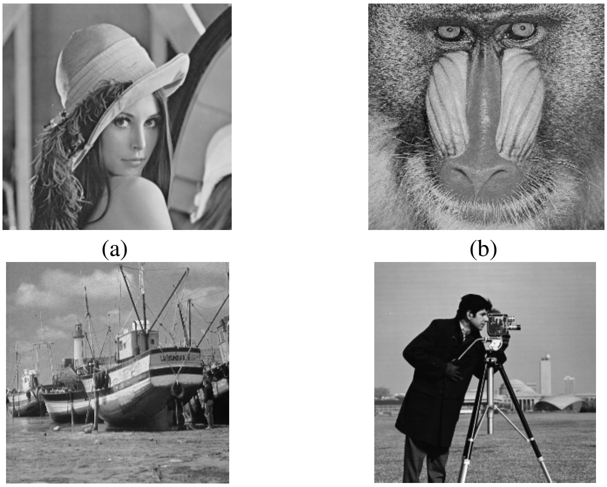
$$\vec{Z_i}(t+1) = \begin{cases} \vec{R_i}(t) & if\ \ f\left(\vec{R_i}(t)\right) \leq f\left(\vec{Z_i}(t)\right) \\ \vec{Z_i}(t) & if\ \ f\left(\vec{R_i}(t)\right) > f\left(\vec{Z_i}(t)\right) \end{cases} \tag{11}$$

where $f(.)$ is the function to be minimized.

5 Experimental Results

DE based multilevel thresholding was performed in MATLAB in a workstation with Intel® Core™2 Duo 2.8 GHz processor. For experimental results four widely popular test images: Lena, Baboon, Boat and Cameraman are chosen (all of 256X256 resolution). The original images are shown in Fig.1. Fig. 2 displays the multilevel thresholded test images. The results are computed for 2, 3 and 4 no of threshold values.

The performance of DE based method is compared with other efficient global optimization techniques like GA and PSO. In case of DE, the following parametric setup is used for all the test images: $Cr = 0.9,\ F = 0.5$. Best possible parametric setup is also maintained for GA and PSO. Results are reported as the mean of the objective functions of 50 independent runs. Each run contains 500 generations. Table 1 shows the required computational time for each levels of "Lena". Results indicate that the computation time of DE is much lesser than the other algorithms and exhaustive search method. Results of mean of the objective function value (f_{mean}) and its standard deviation (f_{std}) for "Lena" image are reported in table 2 which clearly establish the robustness of DE over PSO and GA algorithms for the image segmentation application.



(a) (b) (c) (d)

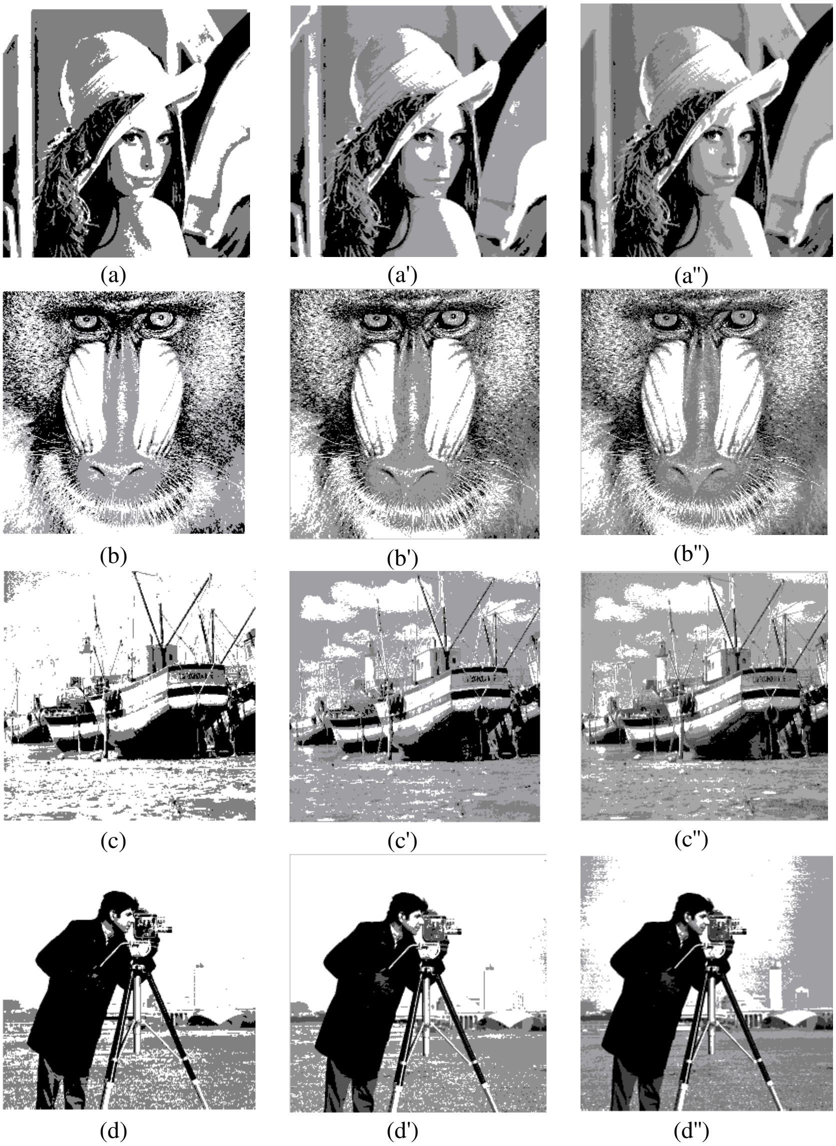
Fig. 1. Original Test Images (a) Lena (b) Baboon (c) Boat (d)Cameraman

Table 1. Average Computational time in Seconds for "Lena" (for 500 generations in one run)

L	DE	PSO	GA	Exhaustive Search
2	**2.4037**	2.7419	3.7186	3.56938 sec.
3	**3.6631**	5.6065	5.7996	> 1 hour
4	**4.1460**	6.5032	8.2662	> 1 day

Different threshold values of test images, obtained by using meta-heuristics, are given in Table 3. Experimental results of exhaustive search is avoided because of its times complexity. For convenience, best results are shown in bold letters. For the performance evolution of DE, PSO and GA correlation, PSNR, Mean Structural Similarity Index Measurement (MSSIM) [14] and Universal Image Quality Index (UIQI) [15] between original image and segmented image are measured.

Correlation and PSNR are tabulated in Table 4, whereas MSSIM and UIQI are shown in Table 5. It shows that results are almost similar for lower no. of threshold values, but for higher no threshold DE has an advantage in the most of the cases.



(a) (a') (a'')

(b) (b') (b'')

(c) (c') (c'')

(d) (d') (d'')

Fig. 2. Thresholded images obtained by MCET-DE method [(a), (b), (c), (d) 2-level thresholding, (a'), (b'), (c'), (d') 3-level thresholding, (a''), (b''), (c''), (d'') 4-level thresholding)]

Table 2. Mean objective function value (f_{mean}) and standard deviation (f_{std}) of "Lena"

L	DE		PSO		GA	
	f_{mean}	f_{std}	f_{mean}	f_{std}	f_{mean}	f_{std}
2	**-4.012e+007**	**0.0**	-4.008e+007	3.496e+004	-4.011e+007	3.617e+003
3	**-4.016e+007**	**0.0**	-4.013e+007	3.816e+004	-4.016e+007	21.070e+000
4	**-4.017e+007**	**0.0**	-4.013e+007	4.207e+004	-4.017e+007	1.515e+000

Table 3. Threshold values acquired by using DE, GA, PSO

Image	*L*	Threshold values		
		DE	PSO	GA
Lena	2	84,142	83,142	83,141
	3	75,120,166	77,122,166	75, 120, 165
	4	73, 110, 142, 177	74, 111,137, 176	62,94, 130, 170
Baboon	2	89,144	88,143	89,144
	3	71,111,153	71, 110, 152	70 , 110, 152
	4	59, 95, 128, 162	56, 96,126, 162	61, 96, 128,161
Boat	2	69 ,132	69 ,132	64, 128
	3	58, 113, 163	57, 110, 160	59, 114, 162
	4	48, 88, 130, 166	48, 94, 134, 166	43, 82, 127, 166
Cameraman	2	51,137	51,134	49, 137
	3	31, 84, 144	29,84,137	30, 83, 144
	4	31, 79, 131, 162	29, 69, 122, 158	29, 76, 125, 158

Table 4. Correlation and PSNR of test images

Image	*L*	Correlation			PSNR(dB)		
		DE	PSO	GA	DE	PSO	GA
Lena	2	**0.9204**	0.9200	0.9197	11.9622	**11.9736**	11.8788
	3	0.9480	**0.9483**	0.9478	**15.5203**	15.4761	15.4566
	4	0.9605	**0.9612**	0.9555	**17.0688**	16.8667	16.7030
Baboon	2	**0.9230**	0.9223	0.9229	**12.0785**	12.0231	**12.0785**
	3	**0.9429**	**0.9429**	0.9424	**13.7640**	13.6677	13.6862
	4	0.9466	0.9463	**0.9475**	**15.3353**	15.3270	15.1896
Boat	2	**0.9124**	**0.9124**	0.9107	**9.7301**	**9.7301**	9.4186
	3	0.9325	0.9275	**0.9357**	**15.8764**	15.4506	15.7879
	4	**0.9434**	0.9404	0.9399	17.0920	**17.1636**	17.0427
Cameraman	2	**0.9610**	0.9605	0.9605	**11.3301**	11.1359	11.3207
	3	**0.9657**	0.9637	0.9657	**11.9115**	11.4522	11.9091
	4	0.9584	0.9572	**0.9612**	**14.2777**	13.5395	13.5476

Table 5. MSSIM and UIQI of test images

Image	No. of Levels	MSSIM			UIQI		
		DE	PSO	GA	DE	PSO	GA
Lena	2	0.5181	0.5201	**0.5210**	0.7550	**0.7551**	0.7520
	3	**0.6144**	0.6085	0.6122	**0.8645**	0.8642	0.8631
	4	**0.6489**	0.6487	0.6609	**0.9032**	0.9005	0.8911
Baboon	2	**0.5435**	0.5429	**0.5435**	**0.7294**	0.7278	**0.7294**
	3	**0.6436**	0.6406	0.6423	**0.7997**	0.7971	0.7981
	4	**0.7199**	0.7194	0.7143	0.8450	**0.8456**	0.8407
Boat	2	0.4660	0.4660	**0.4754**	**0.6921**	**0.6921**	0.6863
	3	**0.5760**	0.5590	0.5724	**0.8590**	0.8472	0.8563
	4	0.6396	0.6435	**0.6453**	**0.8883**	0.8843	0.8870
Cameraman	2	0.5840	0.5802	**0.5843**	**0.8186**	0.8127	0.8178
	3	0.6138	0.6058	**0.6139**	**0.8369**	0.8230	0.8366
	4	0.6159	0.6403	**0.6412**	**0.8863**	0.8743	0.8734

6 Conclusion

The objective of image thresholding for image segmentation is to obtain better segmentation effect within a short span of time. In this paper we have proposed a scheme based on differential evolution for multiple thresholding using MCET. This technique was applied to various real images and the results demonstrated the efficiency and feasibility. The results are encouraging for further research on complex image segmentation and recognition problems. New variations of DE also can be explored in future to achieve improved performance.

References

[1] Sezgin, M., Sankur, B.: Survey over image thresholding techniques and quantitative performance evaluation. Journal of Electronic Imaging 13(1), 146–165 (2004)

[2] Sahoo, P.K., Wilkins, C., Yeager, J.: Threshold selection using Renyi's entropy. Pattern Recognition 30, 71–84 (1997)

[3] Albuquerque, M., Esquef, I.A., Mello, A.R.G., Albuquerque, M.: Image thresholding using Tsallis entropy. Pattern Recognition Letters 25, 1059–1065 (2004)

[4] Li, C.H., Tam, P.K.S.: An iterative algorithm for minimum cross entropy thresholding. Pattern Recognition Letters 19, 771–776 (1998)

[5] Yin, P.: Multilevel minimum cross entropy threshold selection based on particle swarm optimization. Applied Mathematics and Computation 184, 503–513 (2007)

[6] Tanga, K., Sun, T., Yang, J., Gao, S.: An improved scheme for minimum cross entropy threshold selection based on genetic algorithm. Knowledge-Based Systems (2011)

[7] Hammouche, K., Diaf, M., Siarry, P.: A comparative study of various meta-heuristic techniques applied to the multilevel thresholding problem. Engineering Applications of Artificial Intelligence 23, 676–688 (2010)

[8] 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`

[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)

[10] 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)

[11] Pal, S., Qu, B.Y., Das, S., Suganthan, P.N.: Linear Antenna Arrays Synthesis with Constrained Multi-objective Differential Evolution. In: Progress in Electromagnetics Research, PIER B, vol. 21, pp. 87–111 (2010)

[12] Kullback, S.: Information theory and statistics. Dover, New York (1968)

[13] Pal, N.R.: On Minimum Cross-Entropy Thresholding. Pattern Recognition 29(4), 575–580 (1996)

[14] Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image Quality Assessment: From Error Visibility to Structural Similarity. IEEE Transactions on Image Processing 13(4) (2004)

[15] Wang, Z., Bovik, A.C.: A Universal Image Quality Index. IEEE Signal Processing Letters 9(3), 81–84 (2002)

Tuning of Power System Stabilizer Employing Differential Evolution Optimization Algorithm

Subhransu Sekhar Tripathi and Sidhartha Panda

Department of Electrical and Electronics Engineering, National Institute of Science
and Technology (NIST), Berhampur, Orissa 761 008, India
subhransutripathi@yahoo.com, panda_sidhartha@rediffmail.com

Abstract. In this paper, differential evolution (DE) optimization algorithm is applied to design robust power system stabilizer (PSS). The design problem of the proposed controller is formulated as an optimization problem and DE is employed to search for optimal controller parameters. By minimizing the time-domain based objective function, in which the deviation in the oscillatory rotor speed of the generator is involved; stability performance of the system is improved. The non-linear simulation results are presented under wide range of operating conditions; disturbances at different locations as well as for various fault clearing sequences to show the effectiveness and robustness of the proposed controller and their ability to provide efficient damping of low frequency oscillations.

Keywords: Differential evolution, power system stabilizer, low frequency oscillations, power system stability.

1 Introduction

Low frequency oscillations are observed when large power systems are interconnected by relatively weak tie lines. These oscillations may sustain and grow to cause system separation if no adequate damping is available [1]. Power system stabilizers (PSS) are now routinely used in the industry to damp out oscillations. An appropriate selection of PSS parameters results in satisfactory performance during system disturbances [2]. The problem of PSS parameter tuning is a complex exercise. A number of conventional techniques have been reported in the literature pertaining to design problems of conventional power system stabilizers namely: the eigenvalue assignment, mathematical programming, gradient procedure for optimization and also the modern control theory [3]. Unfortunately, the conventional techniques are time consuming as they are iterative and require heavy computation burden and slow convergence. In addition, the search process is susceptible to be trapped in local minima and the solution obtained may not be optimal [4-7].

Differential Evolution (DE) is a branch of evolutionary algorithms developed by Rainer Stron and Kenneth Price in 1995 [8] is an improved version of GA for faster optimization. DE is a population based direct search algorithm for global optimization capable of handling nondifferentiable, nonlinear and multi-modal objective functions, with few, easily chosen, control parameters. The major advantages of DE are its

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 59–67, 2011.
© Springer-Verlag Berlin Heidelberg 2011

simple structure, ease of implementation and robustness. DE differs from other Evolutionary Algorithms (EA) in the mutation and recombination phases. DE uses weighted differences between solution vectors to change the population whereas in other stochastic techniques such as Genetic Algorithm (GA) and Expert Systems (ES), perturbation occurs in accordance with a random quantity. DE employs a greedy selection process with inherent elitist features. Also it has a minimum number of EA control parameters, which can be tuned effectively [9-10]. It has been reported in the literature that DE is far more efficient and robust compared to PSO and the EA [11]. In view of these advantages, a DE-based approach for design of power system stabilizer is proposed in this paper.

2 System Under Study

The SMIB power system with SSSC controller, as shown in Fig. 1, is considered in this study. The system comprises a generator connected to an infinite bus through a step-up transformer and a SSSC followed by a double circuit transmission line. In the figure T represents the transformer; V_T and V_B are the generator terminal and infinite bus voltage respectively; I is the line current and P_L is the real power flow in the transmission lines. The generator is equipped with hydraulic turbine & governor (HTG), excitation system and a power system stabilizer.

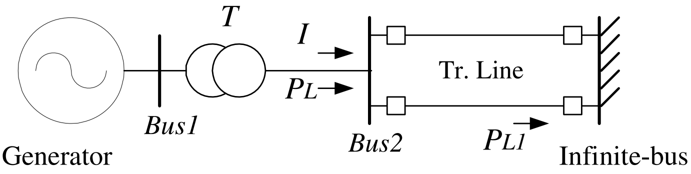


Fig. 1. Single-machine infinite-bus power system

3 The Proposed Approach

3.1 Structure of the Power System Stabilizer

The structure of PSS, to modulate the excitation voltage is shown in Fig. 2. The structure consists of a sensor, a gain block with gain K_P, a signal washout block and two-stage phase compensation blocks as shown in Fig. 2. The input signal of the proposed controller is the speed deviation ($\Delta\omega$), and the output is the stabilizing signal V_S which is added to the reference excitation system voltage. The signal washout block serves as a high-pass filter, with the time constant T_W, high enough to allow signals associated with oscillations in input signal to pass unchanged. From the viewpoint of the washout function, the value of T_W is not critical and may be in the range of 1 to 20 seconds [1]. The phase compensation block (time constants T_1, T_2 and T_3, T_4) provides the appropriate phase lead characteristics to compensate for the phase lag between input and the output signals.

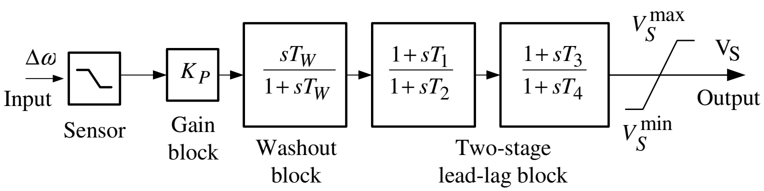


Fig. 2. Structure of power system stabilizer

3.2 Problem Formulation

\In case of lead-lag structured PSS, the sensor and the washout time constants are usually specified. In the present study, a sensor time constant T_{SN} = 15 ms and washout time constant T_W =10s are used. The controller gain K_P and the time constants T_1, T_2, T_3 and T_4 are to be determined.

In the present study, an integral time absolute error of the speed deviations is taken as the objective function expressed as follows:

$$J = \int_{t=0}^{t=t_{sim}} [\ |\Delta\omega|\ ]\cdot t.dt \tag{1}$$

In the above equations, Δω denotes the rotor speed deviation for a set of controller parameters (note that here the controller parameters represent the parameters to be optimized; K_P, T_1, T_2, T_3 and T_4; the parameters of the PSS), and t_{sim} is the time range of the simulation. For objective function calculation, the time-domain simulation of the power system model is carried out for the simulation period. It is aimed to minimize this objective function in order to improve the system response in terms of the settling time and overshoots.

4 Overview of Differential Evolution Optimization Algorithm

Differential Evolution (DE) algorithm is a stochastic, population-based optimization algorithm introduced by Storn and Price in 1996 [8]. DE works with two populations; old generation and new generation of the same population. The size of the population is adjusted by the parameter N_P. The population consists of real valued vectors with dimension D that equals the number of design parameters/control variables. The population is randomly initialized within the initial parameter bounds. The optimization process is conducted by means of three main operations: mutation, crossover and selection. In each generation, individuals of the current population become target vectors. For each target vector, the mutation operation produces a mutant vector, by adding the weighted difference between two randomly chosen vectors to a third vector. The crossover operation generates a new vector, called trial vector, by mixing the parameters of the mutant vector with those of the target vector. If the trial vector obtains a better fitness value than the target vector, then the trial vector replaces the target vector in the next generation. The evolutionary operators are described below [3, 5];

4.1 Inilization

Chromosome For each parameter *j* with lower bound X_j^L and upper bound X_j^U, initial parameter values are usually randomly selected uniformly in the interval $[X_j^L, X_j^U]$.

4.2 Mutation

For a given parameter vector $X_{i,G}$, three vectors ($X_{r1,G}$ $X_{r2,G}$ $X_{r3,G}$) are randomly selected such that the *indices i, r1, r2* and *r3* are distinct. A donor vector $V_{i,G+1}$ is created by adding the weighted difference between the two vectors to the third vector as:

$$V_{i,G+1} = X_{r1,G} + F.(X_{r2,G} - X_{r3,G}) \tag{2}$$

Where *F* is a constant from (0, 2).

4.3 Crossover

The Three parents are selected for crossover and the child is a perturbation of one of them. The trial vector $U_{i,G+1}$ is developed from the elements of the target vector ($X_{i,G}$) and the elements of the donor vector ($X_{i,G}$). Elements of the donor vector enters the trial vector with probability *CR* as:

$$U_{j,i,G+1} = \begin{cases} V_{j,i,G+1} & if \quad rand_{j,i} \le CR \quad or \quad j = I_{rand} \\ X_{j,i,G+1} & if \quad rand_{j,i} > CR \quad or \quad j \ne I_{rand} \end{cases} \tag{3}$$

With $rand_{j,i}$ ~ U (0,1), I_{rand} is a random integer from (1,2,….D) where *D* is the solution's dimension i.e number of control variables. I_{rand} ensures that $V_{i,G+1} \ne X_{i,G}$.

4.4 Selection

An The target vector $X_{i,G}$ is compared with the trial vector $V_{i,G+1}$ and the one with the better fitness value is admitted to the next generation. The selection operation in DE can be represented by the following equation:

$$X_{i,G+1} = \begin{cases} U_{i,G+1} & if \ f(U_{i,G+1}) < f(X_{i,G}) \\ X_{i,G} & otherwise. \end{cases} \tag{4}$$

where $i \in [1, N_P]$.

5 Results and Discussions

The SimPowerSystems (SPS) toolbox is used for all simulations and SSSC-based damping controller design [12]. SPS is a MATLAB-based modern design tool that allows scientists and engineers to rapidly and easily build models to simulate power systems using Simulink environment. In order to optimally tune the parameters of the PSS, as well as to assess its performance, the model of example power system shown in Fig. 1 is developed using SPS blockset.

5.1 Application of DE

For the purpose of optimization of Eq. (4), DE is employed. Implementation of DE requires the determination of six fundamental issues: DE step size function, crossover probability, the number of population, initialization, termination and evaluation function. Generally DE step size (F) varies in the interval (0, 2). A good initial guess to F is in the interval (0.5, 1). Crossover probability (CR) constants are generally chosen from the interval (0.5, 1). If the parameter is co-related, then high value of CR work better, the reverse is true for no correlation [8]. In the present study, a population size of N_P=20, generation number G=200, step size F=0.8 and crossover probability of CR =0.8 have been used [3, 5]. Optimization is terminated by the prespecified number of generations for DE. Simulations were conducted on a Pentium 4, 3 GHz, 504 MB RAM computer, in the MATLAB 7.0.1 environment. The solver options used in the paper are, Variable step type, ode23s (stiff/Mod. Rosenbroc) solver, with a maximum time step of one cycle of the fundamental frequency. The optimization processes is run 20 times and the best values of PSS parameters obtained by the DE algorithm among the 20 runs is given below:

$$K_P=20.4573,\ T_1=0.2000,\ T_2=0.1500,\ T_3=0.2000,\ T_4=0.0280$$

5.2 Simulation Results

The controllers are designed at nominal operating conditions for the system subjected to one particular severe disturbance (3-phase fault). To show the robustness of the proposed design approach, different operating conditions and contingencies are considered for the system with and without controller. In all cases, the optimized parameters obtained for the nominal operating condition given above are used as the controller parameters. Also, the simulation results are compared with a conventional power system stabilizer [1]. Three different operating conditions (nominal, light and heavy) are considered and simulation studies are carried out under different fault disturbances and fault clearing sequences. The response without controller is shown with dotted lines with legend 'NC'; the response with conventional power system stabilizer is shown with dashed line with legend 'CPSS' and the response with proposed DE optimized PSS is shown with solid lines with legend 'DEPSS'.

A 3-cycle 3-phase fault is applied at the middle of one transmission line at the nominal operating conditions (P_e = 0.75 pu, $\delta_0 = 45.37^0$). The fault is cleared after 3-cycles and the original system is restored. The system response for the above contingency is shown in Figs. 3-5. It is also clear from Figs. that, the proposed DE

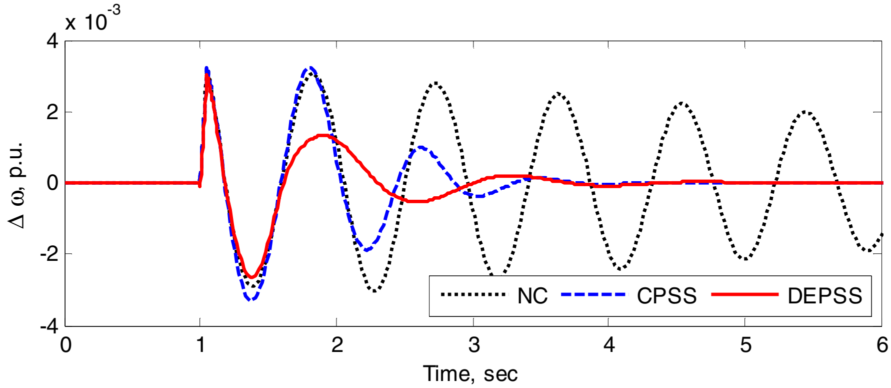


Fig. 3. Speed deviation response of for 3-cycle 3-phase fault at middle of transmission line with nominal loading condition

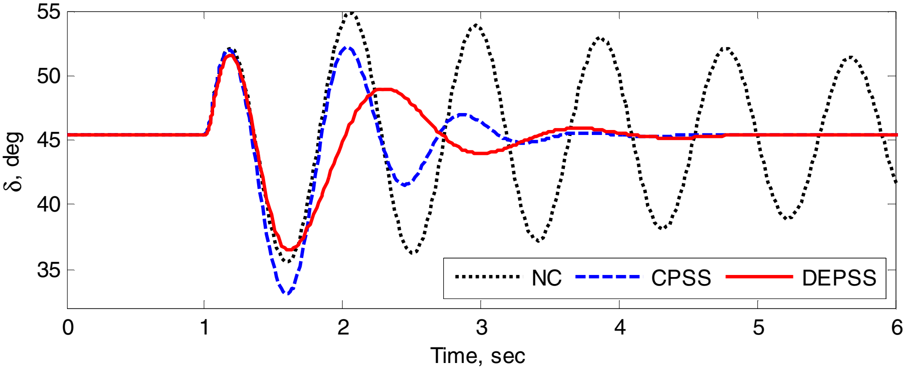


Fig. 4. Power angle response of for 3-cycle 3-phase fault at middle of transmission line with nominal loading condition

optimized PSS outperform the conventional PSS from dynamic performance point of view. The power system oscillations are quickly damped out with the application of proposed PSS.

To test the robustness of the controller to operating condition and fault clearing sequence, the generator loading is changed to light loading condition ($P_e = 0.5$ pu, $\delta_0 = 22.9^0$), and a 3-cycle, 3-phase fault is applied at Bus2. The fault is cleared by opening both the lines. The lines are reclosed after 3-cycles and original system is restored. The system response is shown in Fig. 6 which shows the robustness of proposed PSS to operating conditions and fault clearing sequence.

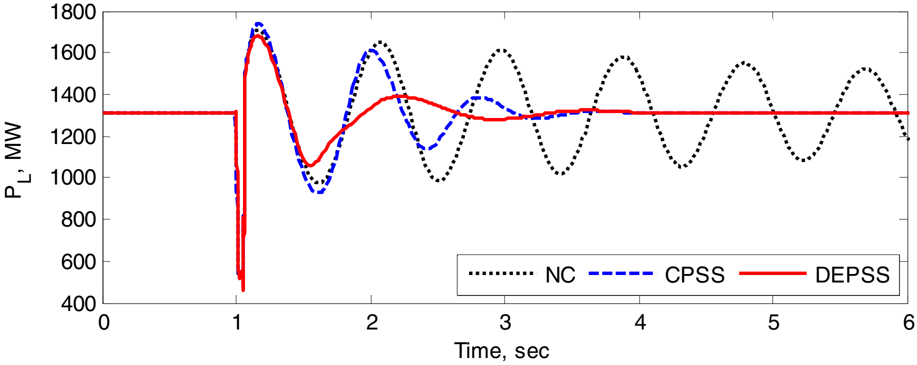


Fig. 5. Tie-line power flow response of for 3-cycle 3-phase fault at middle of transmission line with nominal loading condition

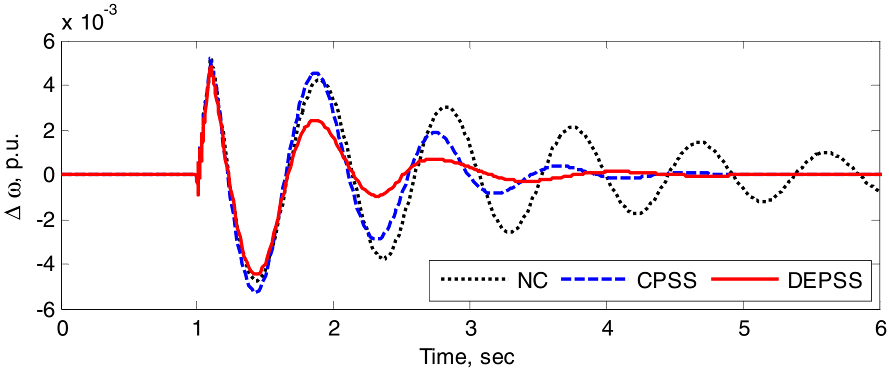


Fig. 6. Speed deviation response of for 3-cycle 3-phase fault at bus-2 cleared by both line outage with light loading condition

The effectiveness and robustness of proposed PSS is also verified under heavy loading condition (P_e = 1.0 pu, $\delta_0 = 60.7^0$). A 3-cycle 3-phase fault is applied near the end of one of the transmission line near infinite bus at t=1.0 s. The fault is cleared by opening the faulty line and the line is reclosed after 3-cycles. The system response is shown in Fig. 7. It can be clearly seen from Figs. 13-15 that, for the given operating condition and contingency, the system is unstable without control. Stability of the system is maintained and power system oscillations are effectively damped out with the application of conventional PSS. The proposed PSS provides the best performance and outperform the conventional PSS by minimizing the transient errors and quickly stabilizes the system.

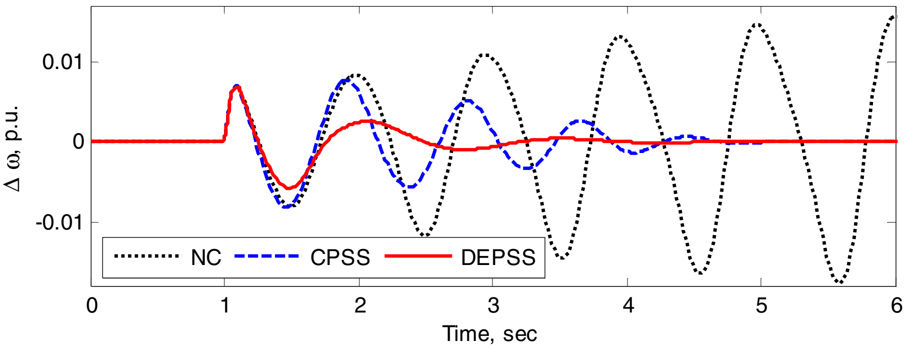


Fig. 7. Speed deviation response of for 3-cycle 3-phase fault near infinite bus at end of one transmission line cleared by one line outage with light heavy condition

6 Conclusion

In this paper, power system stability enhancement by power system stabilizer is presented. For the proposed controller design problem, a non-liner simulation-based objective function to increase the system damping was developed. Then, differential evolution optimization algorithm is implemented to search for the optimal controller parameters. The effectiveness of the proposed controller, for power system stability improvement, is demonstrated by a weakly connected example power system subjected to different severe disturbances. The dynamic performance of proposed PSS has also been compared with a conventionally designed PSS to show its superiority. The non-linear simulation results presented under wide range of operating conditions; disturbances at different locations as well as for various fault clearing sequences, show the effectiveness and robustness of the proposed DE optimized PSS controller and their ability to provide efficient damping of low frequency oscillations..

References

1. Kundur, P.: Power System Stability and Control. McGraw-Hill (1994)
2. Kundur, P., Klein, M., Rogers, G.J., Zywno, M.S.: Application of power system stabilizers for enhancement of overall system stability. IEEE Trans. Power Syst. 4, 614–626 (1989)
3. Panda, S.: Robust coordinated design of multiple and multi-type damping controller using differential evolution algorithm. Int. J. Elect. Power and Energy Syst. 33, 1018–1030 (2011)
4. Panda, S.: Multi-objective PID Controller Tuning for a FACTS-based damping Stabilizer using Non-dominated Sorting Genetic Algorithm-II. Int. J. Elect. Power and Energy Syst., doi:10.1016/j.ijepes.2011.06.002
5. Panda, S.: Differential evolution algorithm for SSSC-based damping controller design considering time delay. Journal of the Franklin Institute, doi:10.1016/j.jfranklin.2011.05.011
6. Panda, S.: Multi-objective evolutionary algorithm for SSSC-based controller design. Electric Power System Research 79, 937–944 (2009)

7. Panda, S., Padhy, N.P.: Comparison of particle swarm optimization and genetic algorithm for FACTS-based controller design. Applied Soft Computing 8, 1418–1427 (2008)
8. Stron, R., Price, K.: Differential evolution – a simple and efficient adaptive scheme for global optimization over continuous spaces. J. Global Optim. 11, 341–359 (1997)
9. Gamperle, R., Muller, S.D., Koumoutsakos, P.: A parameter study for differential evolution. In: WSEAS Int. Conf. on Adv. Intell. Syst. Fuzzy Syst. Evolut. Comput., pp. 293–298 (2002)
10. Zaharie, D.: Critical values for the control parameters of differential evolution algorithms. In: Proceedings of the Eighth International Conference on Soft Computing, pp. 62–67 (2002)
11. Vesterstrom, J., Thomsen, R.: A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems. In: Proceedings of IEEE Congress on Evolut. Comput., CEC 2004, vol. 2, pp. 1980–1987 (2004)
12. SimPowerSystems 5.2.1, `http://www.mathworks.com/products/simpower/`

Logistic Map Adaptive Differential Evolution for Optimal Capacitor Placement and Sizing

Kamal K. Mandal[1,*], Bidishna Bhattacharya[2],
Bhimsen Tudu[1], and Niladri Chakraborty[1]

[1] Jadavpur University, Department of Power Engineering, Kolkata-700098, India
`kkm567@yahoo.co.in`
[2] Techno India, Salt lake, Kolkata, India

Abstract. This paper presents a new adaptive differential evolution technique based on logistic map for optimal distribution placement and sizing. The parameters of differential evolution that need to be selected by the user are the key factors for successful operation DE. Choosing suitable values of parameters are difficult for DE, which is usually a problem-dependent task. Unfortunately, there is no fix rule for selection of parameters. The trial-and-error method adopted generally for tuning the parameters in DE requires multiple optimization runs. Even this method can not guarantee optimal results every time and sometimes it may lead to premature convergence. The proposed method combines differential evolution with chaos theory for self adaptation of DE parameters. The performance of the proposed method is demonstrated on a sample test system. It is seen that the proposed method can avoid premature convergence and provides better convergence characteristics. The results obtained by the proposed methods are compared with other methods. The results show that the proposed technique is capable of producing comparable results.

Keywords: Capacitor placement, Loss reduction, Voltage profile, distribution systems, chaos theory.

1 Introduction

Capacitor placement plays a very important role in distribution system planning and operation. Capacitors have been very commonly used in distribution systems to provide reactive power compensation. They are used to reduce power losses, to improve power factor and to maintain voltage profile with acceptable limits. The objective in capacitor placement problem is to minimize system losses while satisfying various operating constraints under a certain load pattern.

Significant research activity on optimal capacitor placement can be traced back to the 1950's when the famous "two-thirds" rule was developed. According to this rule, a capacitor rated 2⁄3 of the total peak reactive demands needs to be installed at a distance of 2⁄3 along the total feeder length away from the substation for an optimal loss reduction. This rule is still being used as a recommended rule of thumb by many

---

[*] Corresponding author.

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 68–76, 2011.
© Springer-Verlag Berlin Heidelberg 2011

utilities [1]. Lee el al [2] objectively criticized the rule and it was shown that it can be misleading in terms of results accuracy. Grainger et al, [3] presented some better methods to solve the problem using shunt as well fixed and switched capacitor assuming non-uniform load profile. The optimal capacitor placement problem is a complex combinatorial optimization problem and several optimization techniques and algorithm have been applied over the years to solve it. These methods include dynamic programming [4], heuristic numerical algorithm [5], genetic algorithm [6] fuzzy-reasoning method [7], particle swarm optimization technique [8]. Recently, a new method based on plant growth algorithm was proposed by Wang et al [9].

Differential Evolution (DE) is one of the most recent population-based stochastic evolutionary optimization techniques. Storn and Price first proposed DE in 1995 [10] as a heuristic method for minimizing nonlinear and non-differentiable continuous space functions. Several This paper proposes an adaptive differential evolution (DE) technique using chaos theory to solve the problem of distribution capacitor planning. The feasibility of the proposed method is verified on a sample test system. The results have been compared with other evolutionary methods and it is found that it can produce comparable results.

2 Problem Formulation

The objective of capacitor placement problem is to minimize the total annual cost of the system while satisfying some operating constraints under a certain load pattern. The mathematical model of optimal capacitor placement problem can expressed as follows:

$$\min F = \min(COST) \tag{1}$$

where COST includes cost of power loss and capacitor placement.

The voltage magnitude at each bus must be maintained within its limits and is expressed as

$$V_{\min} \le |V_i| \le V_{\max} \tag{2}$$

Where $|V_i|$ is the voltage magnitude of i th bus i, $V_{\min}$ and $V_{\max}$ are the minimum and maximum bus voltage limits respectively.

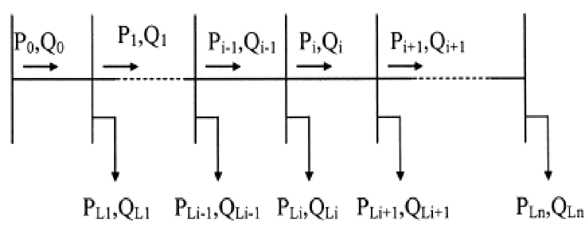


Fig. 1. Single -line diagram of a main feeder

A set of simplified feeder-line flow formulation is assumed for simplicity. Considering the one-line diagram shown depicted in Fig.1, the following set of equations may be used for power flow calculation [Sundharajan and Pahwa 1994].

$$P_{i+1} = P_i - P_{Li+1} - R_{i,i+1} \times \frac{\left(P_i^2 + Q_i^2\right)}{|V_i|^2} \tag{3}$$

$$Q_{i+1} = Q_i - Q_{Li+1} - X_{i,i+1} \times \frac{\left(P_i^2 + Q_i^2\right)}{|V_i|^2} \tag{4}$$

$$|V_{i+1}|^2 = |V_i|^2 - 2\left(R_{i,i+1}.P_i + X_{i,i+1}.Q_i\right) + \left(R_{i,i+1}^2 + X_{i,i+1}^2\right)\frac{\left(P_i^2 + Q_i^2\right)}{|V_i|^2} \tag{5}$$

where P_i and Q_i are the real and reactive powers flowing out of i th bus respectively. P_{Li} , Q_{Li} are the real and reactive load powers at the i th bus respectively. The resistance and reactance of the line section between buses i and i+1 are denoted by $R_{i,i+1}$ and $X_{i,i+1}$ respectively. The power loss of the line section connecting buses i and i + 1 can be calculated as

$$P_{Loss}(i,i+1) = R_{i,i+1} \times \frac{\left(P_i^2 + Q_i^2\right)}{|V_i|^2} \tag{6}$$

The total power loss of the feeder $P_{T,Loss}$ may then be determined by summing up the losses of all line sections of the feeder. The loss is given by

$$P_{T,Loss} = \sum_{i=0}^{n-1} P_{Loss}(i,i+1) \tag{7}$$

The principle of placing compensating capacitor along distribution feeders is to lower the total power loss and transport the bus voltages within their specified limits while minimizing the total cost. Considering the practical capacitors, there exists a finite number of standard sizes which are integer multiples of the smallest size Q_0^c . Besides, the cost per kVAr varies from one size to another. In general, capacitors of larger size have lower unit prices. The available capacitor size is usually limited to

$$Q_c^{\max} = LQ_0^c \tag{8}$$

where L is an integer. Therefore, for each installation location, there are L capacitor sizes $\left\{Q_0^c, 2Q_0^c, ..., LQ_0^c\right\}$ available. Let $\left\{K_1^c, K_2^c, ..., K_L^c\right\}$ be their corresponding equivalent annual cost per kVAr. Therefore, the total annual cost function due to capacitor placement and power loss may be found as

$$COST = K_P P_{T,Loss} + \sum_{i=1}^{n} K_i^c Q_i^c \tag{9}$$

where K_P is the equivalent annual cost per unit of power loss in \$/ (kW – year) and, i = 1, 2, . . . , n are the indices of buses selected for compensation. The bus reactive compensation power is limited to

$$Q_i^c \le \sum_{i=1}^{n} Q_{Li} \tag{10}$$

3 Overview of Differential Evolution

DE or Differential Evolution belongs to the class of evolutionary algorithms [10] that include Evolution Strategies (ES) and conventional genetic algorithms (GA).

DE offers several strategies for optimization [11]. The version used here is the DE/*rand*/1/*bin,* which is described by the following steps.

3.1 Initialization

The optimization process in DE is carried with four basic operations: initialization, mutation, crossover and selection. The algorithm starts by creating a population vector P of size N_P composed of individuals that evolve over G generations. Each individual X_i is a vector that contains as many elements as the problem decision variable. The population size N_P is an algorithm control parameter selected by the user. Thus,

$$P^{(G)} = \left\lfloor X_i^{(G)}, \ldots\ldots\ldots, X_{N_P}^{(G)} \right\rfloor \tag{11}$$

$$X_i^{(G)} = \left[X_{1,i}^{(G)}, \ldots\ldots\ldots\ldots X_{D,i}^{(G)} \right]^T \qquad i = 1, \ldots\ldots\ldots\ldots N_P \tag{12}$$

The initial population is chosen randomly in order to cover the entire searching region uniformly. A uniform probability distribution for all random variables is assumed in the following as

$$X_{j,i}^{(0)} = X_j^{\min} + \sigma_j \left(X_j^{\max} - X_j^{\min} \right) \tag{13}$$

where $i = 1, \ldots\ldots\ldots\ldots\ldots\ldots N_P$ and $j = 1, \ldots\ldots\ldots\ldots\ldots\ldots D$;

Here D is the number of decision or control variables, $X_j^{\min}$ and $X_j^{\max}$ are the lower and upper limits of the j the decision variables and $\sigma_j \in [0, 1]$ is a uniformly distributed random number generated anew for each value of j. $X_{j,i}^{(0)}$ is the j th parameter of the i th individual of the initial population.

3.2 Mutation Operation

Several strategies of mutation have been introduced in the literature of DE. The essential ingredient in the mutation operation is the vector difference. The mutation operator creates mutant vectors $\left(V_i \right)$ by perturbing a randomly selected vector $\left(X_k \right)$ with the difference of two other randomly selected vectors $\left(X_l \text{ and } X_m \right)$ according to:

$$V_i^{(G)} = X_k^{(G)} + f_M \left(X_l^{(G)} - X_m^{(G)} \right) \tag{14}$$

where X_k, X_l and X_m are randomly chosen vectors $\in \left[1, \ldots\ldots, N_P \right]$ and $k \neq l \neq m \neq i$. The mutation factor f_M that lies within [0, 2] is a user chosen parameter used to control the perturbation size in the mutation operator and to avoid search stagnation.

3.3 Crossover Operation

In order to extend further diversity in the searching process, crossover operation is performed. The crossover operation generates trial vectors $\left(U_i \right)$ by mixing the parameter of the mutant vectors with the target vectors. For each mutant vector, an index $q \in \left[1, \ldots\ldots N_P \right]$ is chosen randomly using a uniform distribution and trial vectors are generated according to:

$$U_{j,i}^{(G)} = \begin{cases} V_{j,i}^{(G)} & , \text{ if } \eta_j \leq C_R \text{ or } j = q \\ X_{j,i}^{(G)} & , \text{ otherwise} \end{cases} \tag{15}$$

where $i = 1, \ldots\ldots, N_P$ and $j = 1, \ldots\ldots, D$; η_j is a uniformly distributed random number within [0, 1] generated anew for each value of *j*. The crossover factor $C_R \in \left[0, 1 \right]$ is a user chosen parameter that controls the diversity of the population. $X_{j,i}^{(G)}$, $V_{j,i}^{(G)}$ and $U_{j,i}^{(G)}$ are the *j* th parameter of the *i* th target vector, mutant vector and trial vector at generation *G* respectively.

3.4 Selection Operation

Selection is the operation through which better offspring are generated. The evaluation (fitness) function of an offspring is compared to that of its parent. Thus, if f denotes the cost (fitness) function under optimization (minimization), then

$$X_i^{(G+1)} = \begin{cases} U_i^{(G)} , & \text{if } f\left(U_i^{(G)}\right) \leq f\left(X_i^{(G)}\right) \\ X_i^{(G)} , & \text{otherwise} \end{cases} \tag{16}$$

The optimization process is repeated for several generations. The iterative process of mutation, crossover and selection on the population will continue until a user-specified stopping criterion, normally, the maximum number of generations allowed, is met.

4 Hybrid Differential Evolution Using Chaos Theory

Optimization algorithms based on chaos theory are stochastic search methodologies and are different from the existing evolutionary algorithms. Evolutionary algorithms use the concepts of bio-inspired genetics and natural evolution. On the other hand, optimization techniques using chaos theory based on ergodicity, stochastic properties, and irregularity [12]. Chaotic sequences display an unpredictable long-term behavior due to their sensitiveness to initial conditions. This feature can be utilized to track the chaotic variable as it travels ergodically over the searching space. Crossover ratio and mutation factors are the two important user defined parameters and successful operation of differential evolution is heavily dependent on these two parameters. This paper utilizes chaotic sequence for automatic adjustment of DE parameters. This helps to escape from local minima and improves global convergence.

One of the simplest dynamic systems evidencing chaotic behavior is the iterator called the logistic map [13] and can be described by the following equation.

$$y(t) = \mu \cdot y(t-1) \cdot [1 - y(t-1)] \tag{17}$$

Where t is the sample and μ is control parameter, $0 \le \mu \le 4$.

The behavior of the system described by (17) is greatly changed with the variation of μ. The value of μ determines whether y stabilizes at a constant size, oscillates between a limited sequence of sizes, or behaves chaotically in an unpredictable pattern. Equation (17) is deterministic displaying chaotic dynamics when $\mu = 4$ and $y(0) \notin \{0, 0.25, 0.5, 0.75, 1\}$. In this case, y$(t)$ is distributed in the range of (0,1) provided the initial . $y(0) \in (0,1)$.

The values of the parameters mutation factor (f_M) and cross over ratio (C_R) can be modified using (16) as follows:

$$f_m(G) = \mu \cdot f_m(G-1) \cdot [1 - f_m(G-1)] \tag{18}$$

$$C_R(G) = \mu \cdot C_R(G-1) \cdot [1 - C_R(G-1)] \tag{19}$$

where G is the current iteration number.

5 Structure of Solutions

In this section, an algorithm based on a novel hybrid DE for optimal solution of capacitor placement problem is described. For any population based algorithm like PSO, the representation of individuals and their elements is very important. For the present problem, it is the candidate bus where the capacitor is to be placed. The algorithm starts with the initialization process. Let $P^{(0)} = \left[X_1^{(0)}, X_2^{(0)}, \ldots\ldots X_k^{(0)}, \ldots\ldots, X_{N_P}^{(0)}\right]$ be the initial population of N_p number of particles. For a system of n number of candidate buses, position of k th individual is of n-dimension and can be represented by

$$X_k^{(0)} = \left[Q_{k1}^{c(0)}, Q_{k2}^{c(0)}, \cdots\cdots Q_{kj}^{c(0)}, \cdots\cdots, Q_{kn}^{c(0)} \right] \tag{20}$$

The element $Q_{kj}^{c(0)}$ represents a randomly selected capacitance from the available size satisfying the constraints given by (8).

6 Simulation Results

The proposed algorithm was implemented using in house Matlab code on 3.0 MHz, 2.0 GB RAM PC. To demonstrate the effectiveness and feasibility of the proposed algorithm, it was applied on a sample test system. A population size of 30 and maximum iteration number of 300 was chosen after 50 trial runs for optimal solutions.

The test system [7] under consideration consists of a 23 kV, 9 section feeder. Feeder impedance data, three phase load, available capacitor size with their corresponding annual cost and other data are taken from [7] and not shown here due to page limitation. The equivalent unit cost per unit of power loss considered for the present problem is \$168/(kW-year). The limits on bus voltages are as follows:

$$V_{min} = 0.90\, p.u.$$
$$V_{max} = 1.10\, p.u.$$

It is considered that all the buses were available for compensation. The annual costs, system power loss both before and after compensation, capacitor addition at the desired location are shown in Table 1.

It is seen from Table 1 that voltage profile for all the buses are within the system limits. The annual cost is \$114,757 while the system power loss is 0.6731 MW in comparison with uncompensated cases where the annual cost is \$131,675 and power loss is 0.7837 MW. The convergence characteristic for cost is shown in Fig.2.

Table 1. Results including Voltage profile, Annual cost, Capacitor and Power Loss

Bus No.	Uncompensated Voltage (p.u)	Placed (Qc) (kVar)	Compensated Voltage (p.u)
0	1	0	1.0000
1	0.9929	0	1.0000
2	0.9874	4050	1.0050
3	0.9634	1200	0.9934
4	0.9619	1650	0.9832
5	0.9480	0	0.9614
6	0.9072	1200	0.9554
7	0.8890	0	0.9406
8	0.8587	450	0.9177
9	0.8375	450	0.9011
Total cap. size (Mvar)			9.00
Total Loss (MW)	0.7837		0.6731
Annual cost in (\$/year)	131,675		114,757
CPU time (sec)		80.23	

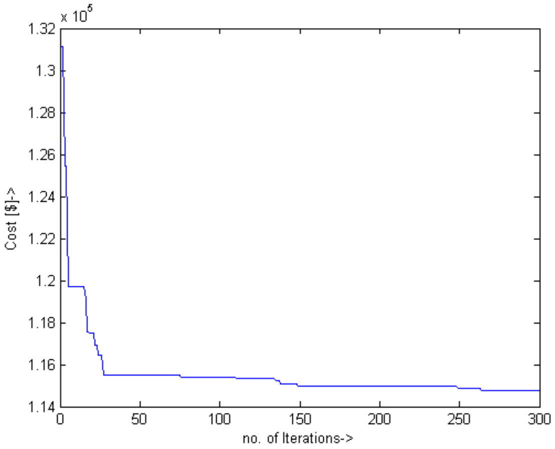


Fig. 2. Convergence Characteristics for Cost

The result is also compared with other methods like fuzzy reasoning [7], DE, GA, ACSA [14] and is shown in Table 2. It is seen from the Table 6, that the proposed method can avoid premature convergence successfully and produce better results.

Table 2. Comparison of results with different methods

	Fuzzy Reasoning	DE	GA	ACSA	Proposed Method
Total Loss (MW)	0.7048	0.6763	0.6766	0.6753	0.6734
Annual cost in ($/year)	119,420	115,471	115,572	115,395	114,757

7 Conclusions

Capacitor placement is one of the important issues in modern day power system operation. The basic objective is to reduce power losses as well as to improve voltage profile. In this paper, an algorithm based on a novel hybrid differential evolution technique has been successfully applied for solving optimal capacitor placement problem to avoid premature convergence. To evaluate the performance of the proposed algorithm, it has been applied on a sample test system. The results obtained by the proposed method have been compared with other population based algorithms like fuzzy reasoning, DE, GA, ACSA. The results show that the proposed algorithm is indeed capable of obtaining good quality solution efficiently in case of capacitor placement problems.

References

[1] Ng, H.N., Salama, M.M.A., Chikhani, A.Y.: Classification of capacitor allocation techniques. IEEE Transactions on Power Delivery 15(1), 387–392 (2000)

[2] Lee, S.H., Grainger, J.J.: Optimum placement of fixed and switched capacitors on primary distribution feeder. IEEE Trans. on Power Apparatus and Systems 100(1), 345–352 (1981)

[3] Grainger, J.J., Lee, S.H.: Capacity release by shunt capacitor placement on distribution feeder: A new voltage-dependent model. IEEE Trans. on Power Apparatus and Systems 101(5), 1236–1244 (1982)

[4] Duran, H.: Optimum number, location, and size of shunt capacitors in radial distribution feeder: A dynamic programming approach. IEEE Trans. on Power Apparatus and Systems 87(9), 1769–1774 (1983)

[5] Baghzouz, Y., Ertem, S.: Shunt capacitor sizing for radial distribution feeders with distorted substation voltages. IEEE Trans. on Power Delivery 5, 650–657 (1990)

[6] Sundharajan, S., Pahwa, A.: Optimal selection of capacitors for radial distribution systems using genetic algorithm. IEEE Trans. Power Systems 9(3), 1499–1507 (1994)

[7] Su, C.T., Tasi, C.C.: A new fuzzy reasoning approach to optimum capacitor allocation for primary distribution systems. In: Proc. 1996 IEEE on Industrial Technology Conference, pp. 237–241 (1996)

[8] Yu, X., Xiong, X., Wu, Y.: A PSO based approach to optimal capacitor placement with harmonic distortion consideration. Electric Power System Research 71, 27–33 (2004)

[9] Wang, C., Cheng, H.Z.: Reactive power optimization by plant growth simulation algorithm. IEEE Trans. on Power Systems 23(1), 119–126 (2008)

[10] Storn, R., Price, K.: Differential Evolution - A simple and efficient adaptive scheme for global optimization over continuous spaces. Journal of Global Optimization 11, 341–359 (1997)

[11] Mallipeddi, R., Suganthan, P.N., Pan, Q.K., Tasgetiren, M.F.: Differential evolution algorithm with ensemble of parameters and mutation strategies. Applied Soft Computing 11(2), 1679–1696 (2011)

[12] Coelho, L.S., Mariani, V.C.: Combining of Chaotic Differential Evolution and Quadratic Programming for Economic Dispatch Optimization with Valve-Point Effect. IEEE Transaction on Power Systems 21(2), 989–996 (2006)

[13] May, R.: Simple mathematical models with very complicated dynamics. Nature 261, 459–467 (1976)

[14] Su, C.T., Chan, C.F., Chiou, J.P.: Capacitor placement in distribution system employing ant colony search algorithm. Electric Components and Systems 33, 931–946 (2005)

Application of an Improved Generalized Differential Evolution Algorithm to Multi-objective Optimization Problems

Subramanian Ramesh[1], Subramanian Kannan[2], and Subramanian Baskar[3]

[1] Arulmigu Kalasalingam College of Engineering, Krishnankoil - 626 126, India
[2] Kalasalingam University, Anand Nagar, Krishnankoil-626 126, India
[3] Thiagarajar College of Engineering, Madurai - 625 015, India
rameshsme@gmail.com, kannaneee@rediffmail.com, sbeee@tce.edu

Abstract. An Improved Multiobjective Generalized Differential Evolution (I-GDE3) approach is proposed in this paper. For maintaining good diversity, the concepts of Simulated Binary Crossover (SBX) based recombination and Dynamic Crowding Distance (DCD) are implemented in GDE3 algorithm. The proposed approach is applied to different sets of classical test problems suggested in the MOEA literature to validate the performance of the I-GDE3. Later, the proposed approach is implemented to Reactive Power Planning (RPP) problem. The objective functions are minimization of combined operating and VAR allocation cost and bus voltage profile improvement. The performance of the proposed approach is tested in standard IEEE 30-bus test systems. The performance of I-GDE3 is compared with respect to multi- objective performance measures namely gamma, spread, minimum spacing and Inverted Generational Distance (IGD). The results show the effectiveness of I-GDE3 and confirm its potential to solve the multi-objective problems.

1 Introduction

Multi-objective optimization problems, unlike a single objective optimization problem, do not necessarily have an optimal solution that minimizes all the multiobjective functions simultaneously. Often, different objectives may conflict each other and the optimal parameters of some objectives usually do not lead to optimality of other objectives (sometimes make them worse).

Evolutionary Algorithms (EAs) can find multiple optimal solutions in single simulation run due to their population approach. The main advantage of evolutionary algorithms (EAs) in solving multi-objective optimization problems is their ability to find multiple Pareto-optimal solutions in one single run. Recently, a number of multiobjective evolutionary algorithms (MOEAs) have been suggested by [5, 25, 26, 28]. Most of these algorithms were developed taking into consideration of two common goals, namely fast convergence to the Pareto-optimal front and good distribution of solutions along the front. Each algorithm employs a unique combination of specific techniques to achieve these goals. MOEA/D [26] decomposes a multiobjective problem into a number of scalar optimization subproblems and

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 77–84, 2011.
© Springer-Verlag Berlin Heidelberg 2011

optimizes them simultaneously. PAES proposed by [13], uses a histogram-like density measure over a hyper-grid division of the objective space. Deb [5] proposed NSGA-II approach which incorporates elitist and crowding approaches.

The differential evolution (DE) algorithm has been found to be successful in single objective optimization problems [21]. Recently there are several attempts to extend the DE to solve multi-objective problems. One approach is presented to optimize train movement by tuning fuzzy membership functions [4]. A Pareto-frontier Differential Evolution algorithm (PDE) to solve multi-objective problem by incorporating Pareto dominance is proposed by [1]. This PDE is also extended [2] with self-adaptive crossover and mutation. An extended DE approach for solving multi-objective optimization problems by incorporating a non-dominated sorting and ranking selection scheme of NSGA-II is presented in [17]. In the literature, the third evolution step of Generalized Differential Evolution (GDE3) has been reported in [14] as an extension of DE for constrained multi-objective optimization. In GDE3, reproduction part of the algorithm is similar to DE [21] and selection and diversity are similar to NSGA-II [5]. Another version of MODE algorithm is found and is applied to industrial styrene reactor [7]. As an extension of this MODE, the concept of elitism is incorporated with the MODE algorithm [6] and it is named as *Elitist* MODE (EMODE) which is applied to Purified Terephthalic Acid (PTA) Oxidation Process. A hybrid version of MODE (H-MODE) which consists of MODE with sequential simplex based local search technique is employed successfully for optimizing the Adiabatic Styrene Reactor and Polyethylene Terephthalate (PET) reactor [8,9]. Wang *et al.* proposes a self-adaptive differential evolution algorithm which incorporates the concept of Pareto dominance to solve multi-objective optimization problem [23].

Recently, the NSGA-II algorithm is used in various power system problems such as economic dispatch [12] and Generation Expansion Planning [11]. Some successful application of EAs to optimal Reactive Power Planning (RPP) has been reported in the literature [15, 3, 27], where minimizing voltage differences have been considered as a multi-objective in addition to cost minimization. The objective of this paper is to solve the multi-objective RPP problem using an Improved Multiobjective Generalized Differential Evolution (I-GDE3) algorithm. This algorithm replaces the crowding distance operator [15] in the original GDE3 algorithm by means of a diversity maintenance strategy which is based on Dynamic Crowding Distance (DCD) proposed by [17]. To validate the performance of the algorithm, performance metrics such as gamma, delta, minimum spacing and inverted generational distance are used.

2 Multi-objective Differential Evolution

Storn and Price proposed an evolutionary algorithm called Differential Evolution (DE) to solve real-parameter optimization problems [21]. DE uses a simple mutation operator based on differences between pairs of solutions (called vectors) with the aim of finding a search direction based on the distribution of solutions in the current population. DE also utilizes a steady-state-like replacement mechanism, where the newly generated offspring (called trial vector) competes only against its corresponding parent (old object vector) and replaces it if the offspring has a higher fitness value.

In order to apply the DE strategy for solving multi-objective optimization problems, the original scheme has to be modified since the solution set of a problem with multiple objectives does not consist of a single solution (as in global optimization). Instead, in multi-objective optimization, the intention is to find a set of different solutions (the so-called Pareto optimal set). Two design aspects should be considered for DE algorithm for extending to multi-objective optimization. The first is to promote diversity among the individuals in the population which can be achieved by means of crowding distance measure and the second is to select the individuals in the population through non-dominated sorting approach proposed by Deb [5]. In the subsequent sections, proposed MODE algorithm is discussed.

2.1 Implementation of I-GDE3 Algorithm

There are different types of MODE algorithm available in the literature. They differ mainly on the type of selection mechanism, mutation strategy and diversity maintenance. In the selection mechanism, MODE can be further classified based on Pareto dominance [1,2,4] and Pareto ranking concepts [6, 7, 8, 9, 14, 17].

In mutation strategy, MODE vary on the type of the criterion to select one of the individuals to be used in the mutation operator (called donor vector), the number of differences computed also in the mutation operator and, finally, in the recombination operator chosen. Several types of mutation strategies were used in MODE namely *DE/current to rand/1/bin* [1, 2, 17], *DE/rand/1/bin* [6, 7, 8, 9, 14] and *DE/best/1/bin* [24]. In diversity maintenance, two major approaches were used. One is based on fitness sharing [4] and the other is based on crowding distance [9, 14].

This paper proposes Simulated Binary Crossover (SBX) as a recombination operator [5] in the MODE algorithm [19]. Besides typical parameters used in EAs (number of individuals and number of iterations), two parameters adopted in I-GDE3 are: crossover index (η_c) and step size (F). The η_c controls the influence of the parents in the generation of the offspring. Higher value of "η_c" means higher probability for creating solutions nearer to the parents and a smaller value of "η_c" allows creating solutions away from the parents. "F" scales the influence of the set of pairs of solutions selected to calculate the mutation value.

It is of great importance for Pareto front with good convergence and diversity characteristics with respect to true Pareto front. A good diversity can give decision-makers more reasonable and efficient selections. The diversity maintenance strategy (DMS) is often realized in the process of population maintenance, which uses a truncation operator to wipe off individuals when the number of non-dominated solutions exceeds population size. In this paper, a DMS which is based on dynamic crowding distance (DCD) is used [16].

2.2 I-GDE3 Algorithm

The following steps can be adopted for the implementation of proposed MODE algorithm.

Step1: Identify the control variables for the problem

Step2: Select the parameters like population size, maximum number of iterations, crossover index (η_c) and mutation parameter (F).

Step3: Generate initial parent population (*P*)

Step4: Evaluate objective functions for initial population

Step5: Set the generation count

Step6: a. Perform *mutation* in the parents to generate mutated parents (Q_m)

b. Perform recombination using SBX crossover (of *P* size), for the entire mutated parents created from *step 6a* (Q_C).

Step7: Perform non-dominated sorting. (i.e., sorting the population according to each objective function value in ascending order of magnitude) for the combined parent and offspring population ($P \cup Q_C$).

Step8: Calculate *DCD* for the combined population based on the algorithm discussed in [16].

Step9: Increment the iteration count and repeat the steps from 6 to 9 until the count reaches the maximum number of iterations.

3 Simulation Results

The I-GDE3 algorithm was implemented using MATLAB version 7.4, on an IBM PC with Pentium dual core processor having 1.86GHz clock speed and 1 GB of RAM.

3.1 Standard Test Problems

Different sets of classical test problems suggested in the MOEA literature are used to validate the performance of the I-GDE3. The results are sensitive to algorithm parameters. Hence, it is required to perform repeated simulations to find suitable values for the parameters. Optimal parameter combinations for different methods are experimentally determined by conducting series of experiments with different parameter settings before conducting actual runs to obtain the results. The crossover index (η_c) is selected between 1 and 10, in steps of 1 and for each η_c performance is analyzed. It is found that $\eta_c = 3$, produces the best results. During simulation, '*F*' is varied in the range 0.1 to 1 in steps of 0.1. It is identified with a step length of 1 is more suitable for better convergence characteristics.

Table 1. Mean and Variance Values of the Convergence (γ) and Divergence (Δ) Metrics for constrained optimization problems & KUR problem

Problem	Algorithms	Convergence		Divergence	
		Mean	Variance	Mean	Variance
OSY	NSGA II	1.5615	0.2232	**0.7482**	0.0882
	I-GDE3	**1.1782**	0.1249	0.7709	0.0752
SRN	NSGA II	0.2397	0.0346	0.3906	0.0452
	I-GDE3	**0.0588**	0.0076	**0.1502**	0.0162
TNK	NSGA II	0.0023	3.59E-4	0.8481	0.1005
	I-GDE3	**0.00145**	2.21E-4	**0.3640**	0.0421
KUR	NSGA II	0.028964	0.000018	**0.4115**	0.000992
	I-GDE3	**0.0021233**	0.00018741	0.4936	0.00513

The performance of the proposed I-GDE3 algorithm is tested in a benchmark test unconstrained (KUR) optimization and constrained optimization problems such as OSY, SRN and TNK [5]. The performance of I-GDE3 is compared with respect to the multi-objective performance measures (namely convergence and divergence metrics) and it is reported in Table 1. The results show the effectiveness of I-GDE3 and confirm its potential to solve the multi-objective RPP problem. All results have been averaged over 30 independent runs. A result with boldface indicates better value obtained. From the Table 1, it is observed that the performance of I-GDE3 is better than NSGA-II in the standard benchmark test problems.

3.2 Implementation of I-GDE3 to RPP Problem

This paper considers a short-term RPP, where the investment is to be performed only once. The first objective function comprises two terms [10, 15]. The first term represents the total cost of energy loss and the second term represents the cost of VAR source installations which has two components, namely, fixed installation cost and cost of the VAR source. The second objective is to minimize the deviations in voltage magnitudes at load bus [3, 27]. In the control variables, the generator bus voltages i.e. Automatic Voltage Regulator operating values are taken as continuous variable, whereas the transformer tap settings and shunt susceptance values are taken as discrete values.

The IEEE *30-bus* test system is considered as a test system the detailed data are given in [22]. The system has 6 generators, 4 transformers and 9 shunt compensators; therefore, the numbers of variables to be optimized are 19. The bus numbers 1, 2, 5, 8, 11, and 13 are generator bus. The lower and upper limits for voltage magnitude of these buses are 0.95 p.u. and 1.1 p.u., and for the remaining bus lower and upper limits for voltage magnitude of bus are 0.95 p.u. and 1.05 p.u. respectively. The transformer tapping are varied between 0.9 and 1.1 p.u. with the step size of 0.0001 and the shunt capacitors have the rating between 0 and 5 MVAR with each capacitor with the step size of 1 MVAR.

The RPP problem is treated as true multi-objective optimization problem where combined operating and investment costs and voltage deviations are optimized simultaneously with the I-GDE3 algorithm.

The NSGA-II and I-GDE3 algorithm are applied to solve the RPP problem and the best pareto-front obtained among 15 simulation runs is represented in fig. 1. In some problems, the true pareto-front may not be available. For that case, the determination of reference pareto-front is needed. In this paper, Covariance Matrix Adapted Evolution Strategy (CMA-ES) technique [18] with conventional weighted sum approach is used as reference pareto-front since for a RPP problem there is no true pareto-front reported in the literature. So the performance metrics have been calculated for NSGA-II, MNSGA-II [20] and I-GDE3 (by comparing with the reference pareto-front generated by CMA-ES).The statistical results of performance metrics such as mean and standard deviation (σ) in 15 simulation runs are represented in Table 2 and the bold numbers indicate the best mean values. From Table 2, it is observed that except for minimum spacing, the performance of I-GDE3 is better than NSGA-II and MNSGA-II with respect to the mean values of metrics.

Table 2. Statistical Comparison of Performance Metrics IEEE *30-bus*

Performance Measures		NSGA-II	I-GDE3	MNSGA-II
Gamma (γ)	Mean	0.1146	**0.1118**	**0.1118**
	σ	0.0854	0.0765	0.1071
Spread (Δ)	Mean	0.0675	**0.0621**	0.0629
	σ	0.0711	0.0638	0.0668
Minimum Spacing (S_m)	Mean	**0.0376**	0.0412	0.0542
	σ	0.0590	0.0503	0.0503
IGD	Mean	0.2311	**0.2281**	0.2291
	σ	0.0362	0.0473	0.0563

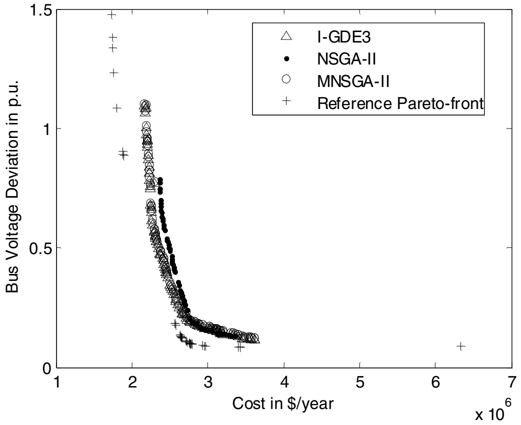


Fig. 1. Comparison of Pareto-Optimal Set for IEEE *30-bus*

4 Conclusion

In this paper, Improved Generalized Differential Evolution (I-GDE3) is applied to standard multi-objective test problem and reactive power planning problem (RPP). In RPP, conflicting objectives such as combined operating and VAR allocation cost minimization and bus voltage profile improvement are considered. The IEEE *30-bus* system is considered as a test system. By using conventional weighted sum method with Covariance Matrix Adapted Evolution Strategy (CMA-ES), a reference pareto-front is generated. The statistical multi-objective performance measures such as gamma, spread, minimum spacing and Inverted Generational Distance are considered for validating the improved performance of I-GDE3. From the simulation results, it was observed that I-GDE3 performs better in most of the performance measures when compared to NSGA-II for both standard test problem and RPP problem. The simulation result clearly shows that I-GDE3 algorithm is certainly more suitable for solving multi-objective RPP problems.

Acknowledgments. The authors gratefully acknowledge the management of Kalasalingam University, Krishnankoil, Thiagarajar College of Engineering, Madurai, Tamilnadu, India. The authors are also thankful to the Department of Science and Technology, New Delhi for its support through the project SR/S4/MS/350/06 to the second author.

References

1. Abbass, H.A., Sarker, R., Newton, C.: PDE: A Pareto-frontier Differential Evolution Approach for Multi-objective Optimization Problems. In: Proc. of IEEE Cong. on Evol. Comp., pp. 971–978 (2001)
2. Abbass, H.A.: The Self-Adaptive Pareto Differential Evolution Algorithm. In: Proc.of IEEE Cong. on Evol. Comp., vol. 1, pp. 831–836 (2002)
3. Abido, M.A., Bakhashwain J.M.: A Novel Multi-objective Evolutionary Algorithm for Reactive Power Dispatch Problem. In: ICECS 2003, pp. 1054–1057 (2003)
4. Chang, C.S., Xu, D.Y., Quek, H.B.: Pareto-optimal set based multi-objective tuning of fuzzy automatic train operation for mass transit system. IEE Proceedings on Electric Power Applications 146(5), 577–583 (1999)
5. Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms, 1st edn. John Wiley & Sons, Ltd., Singapore (2001)
6. Gujarathi, A.M., Babu, B.V.: Improved Multi-Objective Differential Evolution (MODE) Approach for Purified Terephthalic Acid (PTA) Oxidation Process. Materials and Manufacturing Processes 24(3), 303–319 (2009)
7. Gujarathi, A.M., Babu, B.V.: Multi-objective Optimization of Industrial Styrene Reactor: Adiabatic and Pseudo-isothermal Operation. Chem. Engg. Science 65(6), 2009–2026 (2010)
8. Gujarathi, A.M., Babu, B.V.: Optimization of Adiabatic Styrene Reactor: A Hybrid Multi-Objective Differential Evolution (H–MODE) Approach. Industrial & Engineering Chemistry Research 48(24), 11115–11132 (2009)
9. Gujarathi, A.M., Babu, B.V.: Hybrid Multi-objective Differential Evolution (HMODE) for optimization of Polyethylene Terephthalate (PET) reactor. Int. J. of Bio. Insp. Comp. 2(3/4), 213–221 (2010)
10. Hsaio, Y.T., Chaing, H.D., Liu, C.C., Chen, Y.L.: A Computer Package for Optimal Multi-objective VAR Planning in Large Scale Power Systems. IEEE Trans. on Power Syst. 9(2), 668–676 (1994)
11. Kannan, S., Baskar, S., McCalley, J.D., Murugan, P.: Application of NSGA-II Algorithm to Generation Expansion Planning. IEEE Trans. on Power Syst. 24(1), 454–461 (2009)
12. King, R.T.F.A., Rughooputh, H.C.S.: Elitist multiobjective evolutionary algorithm for environmental/economic dispatch. In: The 2003 Congress on Evolutionary Computation, vol. 2, pp. 1108–1114 (2003)
13. Knowles, J., Corne, D.: The Pareto archived evolution strategy: A new baseline algorithm for multi-objective optimization. In: Proceedings of the 1999 Congress on Evolutionary Computation, pp. 98–105 (1999)
14. Kukkonen, S., Lampinen, J.: GDE3: The third evolution step of Generalized Differential Evolution. In: Proceedings of the 2005 Congress on Evolutionary Computation (CEC 2005), pp. 443–450 (2005)
15. Lai, L.L., Ma, J.T.: Application of Evolutionary Programming to Reactive Power Planning – Comparison with Nonlinear Programming Approach. IEEE Trans. on Power Syst. 12(1), 198–206 (1997)

16. Luo, B., Zheng, J., Xie, J., Wu, J.: Dynamic Crowding Distance - A New Diversity Maintenance Strategy for MOEAs. In: ICNC 2008, 4th International Conference Natural Computation, vol. 1, pp. 580–585 (2008)
17. Madavan, N.K.: Multiobjective Optimization Using a Pareto Differential Evolution Approach. In: Proceedings of Congress on Evolutionary Computation (CEC 2002), vol. 2, pp. 1145–1150 (2002)
18. Manoharan, P.S., Kannan, P.S., Baskar, S., Willjuice Iruthayarajan, M., Dhananjeyan, V.: Covariance matrix adapted evolution strategy algorithm-based solution to dynamic economic dispatch problems. Engg. Optimization 41(7), 635–657 (2009)
19. Ramesh, S., Kannan, S., Baskar, S.: An improved generalized differential evolution algorithm for multi-objective reactive power dispatch. Engineering Optimization (2011), doi:10.1080/0305215X.2011.576761
20. Ramesh, S., Kannan, S., Baskar, S.: Multi-objective Evolutionary Algorithm based Reactive Power Dispatch with Thyristor Controlled Series Compensators. In: IEEE PEDES 2010 and Power India, pp. 1–5 (2010)
21. Storn, R., Price, K.: Differential evolution-A simple and Efficient Heuristic for Global Optimization over Continuous Spaces. J. Glob. Optim. 11, 341–359 (1997)
22. University of Washington, Power Systems Test Case Archive, `http://www.ee.washington.edu/research/pstca/`
23. Wang, Y., Wu, L., Yuan, X.: Multi-objective self-adaptive differential evolution with elitist archive and crowding entropy-based diversity measure. Soft Computing 14(3), 193–209 (2010)
24. Xue, F., Sanderson, A.C., Graves, R.J.: Modeling and convergence analysis of a continuous multi-objective differential evolution algorithm. In: 2005 IEEE Congress on Evolutionary Computation (CEC 2005), vol. 1, pp. 228–235 (2005)
25. Zhao, S.Z., Suganthan, P.N.: Two-lbests Based Multi-objective Particle Swarm Optimizer. Engineering Optimization 43(1), 1–17 (2011)
26. Zhao, S.Z., Suganthan, P.N., Zhang, Q.: Decomposition Based Multiobjective Evolutionary Algorithm with an Ensemble of Neighborhood Sizes. IEEE Trans. on Evolutionary Computation (accepted)
27. Zhihuan, L., Yinhong, L., Xianzhong, D.: Non-dominated sorting genetic algorithm-II for robust multi-objective optimal reactive power dispatch. IET Gen., Trans. and Dist. 4(9), 1000–1008 (2010)
28. Zhou, A., Qu, B.Y., Li, H., Zhao, S.Z., Suganthan, P.N., Zhang, Q.: Multiobjective evolutionary algorithms: A survey of the state of the art. Swarm and Evolutionary Computation 1, 32–49 (2011)

Enhanced Discrete Differential Evolution to Determine Optimal Coordination of Directional Overcurrent Relays in a Power System

Joymala Moirangthem[1], Subranshu Sekhar Dash[1],
K.R. Krishnanand[2], and Bijaya Ketan Panigrahi[3]

[1] Department of Electrical and Electronics Engineering, SRM University, Tamil Nadu, India
[2] Multi-Disciplinary Research Cell, Siksha 'O' Anusandhan University, Bhubaneswar, India
[3] Department of Electrical Engineering, Indian Institute of Technology, Delhi, India
joy.mala@gmail.com, munu_dash_2k@yahoo.com,
krishkr09@gmail.com, bkpanigrahi@ee.iitd.ac.in

Abstract. This paper presents an enhanced differential evolution technique to solve the optimal coordination of directional overcurrent relays in a power system. The most vital task when installing directional relays on the system is selecting suitable current and time settings such that their fundamental protective functions are met under the requirements of sensitivity, selectivity, reliability and speed. Coordination of directional over current relays can be formulated as a linear or non-linear problem taking into account the discrete value for the time dial settings and the pickup current settings. The results are compared with different algorithms on a test system and are presented in this paper.

Keywords: Directional overcurrent relays (DOCRs), Enhanced Discrete Differential Evolution Algorithm (EDDEA), Relay coordination, Pickup current settings (Ip), Time dial settings (TDS).

1 Introduction

The basic role of a transmission protection system is to sense faults on lines or at a substation and to rapidly isolate these faults by opening all incoming current paths. It should be very selective in disconnecting the network from service to avoid unnecessary removal of network and also it should reliable. This need has made the protection of system to come up with two types of protection i.e. primary protection and back up protection. The primary relays are designed for speed and minimum disruption of network while back up relay operates more slowly than the primary and affects larger network [1]. The problem of coordination involves computation of time dial setting (TDS) and pick up current (Ip) setting or plug setting. Various techniques have been approached to solve the coordination problem since the manual computation is very tedious and time consuming task. Many researchers report on computing the optimal coordination of directional overcurrent relays by using linear programming techniques such simplex method [2]-[4], dual simplex methods [5]. A detail report has been performed by Birla et al in solving coordination problem by three techniques namely curve-fitting technique, graph theoretical technique and

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 85–93, 2011.
© Springer-Verlag Berlin Heidelberg 2011

optimization technique [6]. In [7] random search technique to achieve proper coordination with acceptable speed of the primary protection. Stochastic search technique like GA and PSO started applying by many researchers to solve the coordination problem. So and Li proposed GA [8] algorithm to compute the optimal setting of DOCR. A modified PSO [9] was proposed by Zeineldin et al with less computation time to determine global optimum value of settings. Differential evolution (DE) is an efficient and powerful population-based stochastic search technique for solving optimization problems over continuous space, which has been widely applied in many scientific and engineering fields [10]. Bashir et al implemented a new hybrid PSO [11] to compute optimal coordination of DOCR considering TMS as continuous parameter and PS as discrete parameter.

This paper is organized as follows: In Section 2, mathematical model formulation for solving the coordination problem is discussed. Section 3 describes about the enhanced differential evolutionary algorithm which have been applied to solve the problem. In Section 4, the proposed method is carried in 8 bus test system and the results are discussed. Finally, conclusions are summarized in Section 5.

2 Problem Formulation

Identifying the relay requirements precisely and unambiguously and also the relay coordination criteria to be adopted are the most important and primary task to be undertaken by the protection engineers to ensure the high efficiency of system protection. Relay characteristics and coordination criteria are summarized below.

2.1 Relay Characteristics

The operating time of the overcurrent relay is a non-linear function of pickup current setting (I_p) and time dial setting (TDS). The pickup current is the minimum current value above which the relay trips. The time dial setting determines the operation time of the relay for each current value given by the typical time characteristic T Vs M. Where, M (multiple of the pickup current) is the ratio of the relay fault current to the pickup current setting, i.e. $M=I_f/I_p$. The relay characteristics function can be represented by equation (1) and is given by equation (2).

$$T_i=f_i(TDS_i,I_{pi},I_{fi}) \tag{1}$$

$$T=\frac{(0.14*TDS)}{(M^{0.02}-1)} \tag{2}$$

Where, TDS_i is the time dial setting of relay R_i, I_{pi} is the pickup current setting of relay R_i and I_{fi} is the current flowing in the relay R_i for a particular fault located in zone k.

2.2 Relay Setting

The calculation of the two settings, TDS (discrete) and I_P (discrete) is the most important part of the overcurrent relay coordination. I_P also correspond to the plug-setting of the relay. Each relay pickup current has lower and upper limit values. The lower limit is the larger of the minimum tap available or a factor times the maximum load current flowing through the relay. The upper limit is the smaller of the fault current. Similarly TDS has also lower and upper limit values based on the relay current-time characteristic. The constraints can be formulated as

$$I_{pi\min} < I_{pi} < I_{pi\max} \tag{3}$$

$$TDS_{i\min} < TDS_i < TDS_{i\max} \tag{4}$$

2.3 Coordination Criteria

\To ensure relay coordination, the operating time of the backup relay should exceed that of its corresponding primary relay for all the faults by a coordination time interval (CTI). The typical value of CTI is 0.2 sec. CTI constraint is illustrated.

$$T_{backup} - T_{primary} \geq CTI \tag{5}$$

Where, T_{backup} is the operating time of backup relay and $T_{primary}$ is the operating time of the primary relay. After considering all these criteria, overcurrent relays coordination problem can be formulated as minimization of

$$J = \sum_{i=1}^{N} T_{pri} \tag{6}$$

Where N is the number of relays operating and T_{pri} is the operating time of the primary relay for near-end fault.

3 Enhanced Differential Evolution

Evolutionary Algorithms (EAs), inspired by the natural evolution of species, have been successfully applied to solve numerous optimization problems in diverse fields. The differential evolution (DE) algorithm, proposed by Storn and Price [12], is a simple yet powerful population-based stochastic search technique, which is an efficient and effective global optimizer in the continuous search domain. DE has been successfully applied in diverse fields such as power systems [13], mechanical engineering [13], communication [13] and pattern recognition [13]. In DE, there exist three crucial control parameters, i.e., population size, scaling factor, and crossover rate, may significantly influence the optimization performance of the DE.

DE algorithm aims at evolving a population of NP D–dimensional parameter vectors, so-called individuals, which encode the candidate solutions towards the global optimum. The parameters of the ith vector for the generation g are given by equation (7).

$$X_{(i)}^{g} = \left\{ x_{(i,1)}^{g}, x_{(i,2)}^{g}, \ldots, x_{(i,j)}^{g}, \ldots, x_{(i,D)}^{g} \right\} \tag{7}$$

$g = 1,2, \ldots, G$ and $i = 1,2, \ldots, N$, where G is the maximum number of generations, N is the population size and D is the dimension of the problem. The initial population should better cover the entire search space as much as possible by uniformly randomizing individuals within the search space constrained by the prescribed minimum and maximum parameter bounds.

$$X^{\min} = \left\{x_1^{\min}, x_2^{\min}, \ldots, x_j^{\min}, \ldots, x_D^{\min}\right\} \tag{8}$$

$$X^{\max} = \left\{x_1^{\max}, x_2^{\max}, \ldots, x_j^{\max}, \ldots, x_D^{\max}\right\} \tag{9}$$

The j^{th} parameter of the i^{th} vector at the first generation is initialized randomly using the equation

$$x_{(i,j)}^{1} = x_j^{\min} + rand_{(i,j)} \times \left(x_j^{\max} - x_j^{\min}\right) \tag{10}$$

Where $rand_{(i,j)}$ represents a uniformly distributed random variable within the range (0,1).

3.1 Mutation Operation

After initialization, DE employs the mutation operation to produce a mutant vector *V* with respect to each individual ***X*,** so-called target vector, in the current population. For each target vector *X* at the generation *g*, its associated mutant vector is can be express as,

$$V_{(i)}^{g} = \left\{v_{(i,1)}^{g}, v_{(i,2)}^{g}, \ldots, v_{(i,j)}^{g}, \ldots, v_{(i,D)}^{g}\right\} \tag{11}$$

It can be generated via certain mutation strategy. For example, the five most frequently used mutation strategies implemented in the DE are listed as follows:

1) “DE/rand/1”:

$$V_{(i)}^{g} = X_{(\alpha)}^{g} + F \times \left(X_{(\beta)}^{g} - X_{(\gamma)}^{g}\right) \tag{12}$$

2) “DE/best/1”:

$$V_{(i)}^{g} = X_{best}^{g} + F \times \left(X_{(\alpha)}^{g} - X_{(\beta)}^{g}\right) \tag{13}$$

3) “DE/rand-to-best/1”:

$$V_{(i)}^{g} = X_{(i)}^{g} + F_1 \times \left(X_{best}^{g} - X_{(i)}^{g}\right) + F_2 \times \left(X_{(\alpha)}^{g} - X_{(\beta)}^{g}\right) \tag{14}$$

4) “DE/best/2”:

$$V_{(i)}^{g} = X_{best}^{g} + F_1 \times \left(X_{(\alpha)}^{g} - X_{(\beta)}^{g}\right) + F_2 \times \left(X_{(\gamma)}^{g} - X_{(\zeta)}^{g}\right) \tag{15}$$

5) “DE/rand/2”:

$$V_{(i)}^{g} = X_{(\alpha)}^{g} + F_1 \times \left(X_{(\beta)}^{g} - X_{(\gamma)}^{g}\right) + F_2 \times \left(X_{(\zeta)}^{g} - X_{(\eta)}^{g}\right) \tag{16}$$

The indices $\alpha, \beta, \gamma, \zeta, \eta$ are mutually exclusive integers randomly generated within the range [1, N], which are also different from the index i. These indices are randomly generated once for each mutant vector for a particular generation g. The scaling factor F, F_1 and F_2 are positive control parameters for scaling the difference vector. $X_{(best)}$ is the best individual vector with the best fitness value in the population at generation g.

3.2 Crossover Operation

After the mutation phase in the generation g, crossover operation defined by equation (18) is used to generate each trial vector represented in equation (17).

$$U_{(i)}^{g} = \left\{u_{(i,1)}^{g}, u_{(i,2)}^{g}, \dots, u_{(i,j)}^{g}, \dots, u_{(i,D)}^{g}\right\} \tag{17}$$

$$u_{(i,j)}^{g} = \begin{Bmatrix} v_{(i,j)}^{g} & if\,(rand_{(i,j)} \le Cr) \quad or \quad j = j_{rand} \\ x_{(i,j)}^{g} & otherwise \end{Bmatrix} \tag{18}$$

The crossover rate Cr is a user-specified constant within the range (0, 1), which controls the fraction of parameter values copied from the mutant vector. j_{rand} is a randomly chosen integer in the range [1, D]. The binomial crossover operator copies the j^{th} parameter of the mutant vector $V_{(i)}$ to the corresponding element in the trial vector $U_{(i)}$ if $rand_{(i,j)} \le Cr$ or $j = j_{\text{rand}}$. Otherwise, it is copied from the corresponding target vector $X_{(i)}$. This ensures that the trial vector will differ from its corresponding target vector by at least one parameter.

3.3 Selection Operation

After the objective function values of all trial vectors are evaluated, a selection operation is performed. The objective function value of each trial vector $f(U_{(i)})$ is compared to that of its corresponding target vector $f(X_{(i)})$ in the current population. If the trial vector has less or equal objective function value than the corresponding target vector, the trial vector will replace the target vector and enter the population of the next generation. Otherwise, the target vector will remain in the population for the next generation. The selection operation can be expressed as follows:

$$X_{(i)}^{g+1} = \begin{Bmatrix} U_{(i)}^{g} & if \quad f\left(U_{(i)}^{g}\right) \le f\left(X_{(i)}^{g}\right) \\ X_{(i)}^{g} & if \quad f\left(U_{(i)}^{g}\right) > f\left(X_{(i)}^{g}\right) \end{Bmatrix} \tag{19}$$

3.4 Enhanced Discrete DE

The differential evolution used in this paper requires operation on discrete values since the decision variables to be optimized (TDS and I_p) are discrete by nature. The input dimensions of the objective function to be optimized (summation of the operating time of all the relays), possess piecewise existential domains. So, before evaluation of the each potential solution vector in DE, the vectors generated are quantized to two digit precision when expressed in engineering notation.

The DE algorithm is enhanced by making the scaling factors F, F_1 and F_2 time varying and stochastic using the equations (20)-(22).

$$F^g = F^* + \lambda \cdot \phi^g \tag{20}$$

$$F_1^g = F_1^* + \lambda_1 \cdot \phi_1^g \tag{21}$$

$$F_2^g = F_2^* + \lambda_2 \cdot \phi_2^g \tag{22}$$

Where F^g values are the time varying enhanced scaling factors F^* values are the constant offsets which act as the mean around which the scaling factors oscillate. The ϕ values represent random values which follow normal distribution function with zero as mean. The λ values are the multipliers which amplify the random values. Since the F^g values act as multipliers of differential vectors, it is often possible for the DE to get stuck at a local optimum. As the domains of the decision variables are discontinuous, it is difficult for the conventional algorithm to perturb the potential solutions so that the decision variables can have values which exist in their feasible domain. The time varying scaling factors ensure that the mutation operation causes necessary variation in the decision variables so that they can jump to other feasible points. This makes the DE enhanced for such a discrete problem.

4 Simulation Results

4.1 System Data

The proposed method for solving the coordination problem was tested on 8 bus test system which is shown in Fig.1. The TDS are assumed to vary from 0.1 to 1.1 and the available pickup current setting is [0.5, 0.6, 0.8, 1, 1.5, and 2.5]. The coordination time interval (CTI) is taken as 0.2.

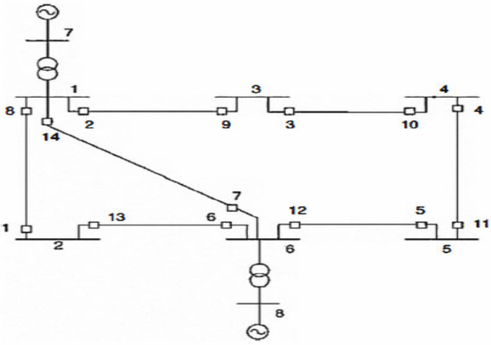


Fig. 1. 8 bus test system

The three phase faults are applied at the near-end of each phase relay. The CT ratios for the pickup current setting for 3 bus system are given Table 1. The primary and back up pairs and also the fault currents are given in Table 2.

Table 1. CT ratio for 8 bus system

Relay No.	CT Ratio	Relay No.	CT Ratio	Relay No.	CT Ratio
1	1200/5	6	1200/5	11	1200/5
2	1200/5	7	800/5	12	1200/5
3	800/5	8	1200/5	13	1200/5
4	1200/5	9	800/5	14	800/5
5	1200/5	10	1200/5		

Table 2. Primary and Backup pair information for 8 bus systems

Relay No.	Primary Fault current (I_f), Amps	Relay No.	Backup Fault current (I_f), Amps	Relay No.	Primary Fault current (I_f), Amps	Relay No.	Backup Fault current (I_f), Amps
13	1.400	1	5.000	10	2.600	11	4.300
3	0.176	2	4.900	6	0.485	12	4.800
4	2.700	3	4.300	5	1.600	13	4.900
12	1.700	4	4.300	1	2.600	14	5.300
11	0.831	5	4.800	14	0.666	2	4.900
8	0.202	6	4.900	14	1.800	6	4.900
11	0.202	7	5.300	12	0.485	7	5.300
2	0.666	8	5.000	7	0.086	8	5.000
1	0.745	9	4.900	7	1.800	12	4.800
9	1.700	10	4.300	2	0.086	14	5.300

4.2 Implementation of Enhanced Discrete Differential Evolution

DE applied to the relay coordination problem formulation and was coded in MATLAB with total number of variables 28 and population size of 20 particles. The maximum number of generation count used is 1000. The enhanced discrete DE performs best with the last strategy in DE. The scaling factor F_1^* and F_2^* are taken to be as 0.5 and 0.3. Both λ_1 and λ_2 were taken as 0.8. The normal distribution function (represented by ϕ_1^g and ϕ_2^g) generates positive and negative values in continuous domain with mean as zero. The crossover rate used is 0.8. The optimal time dial setting and pick up current setting of each overcurrent relay for the test systems are calculated. The optimal values are shown in Table 3.

The results obtained by using Enhanced Discrete DE are compared with other algorithms and found that the Enhanced DE algorithm gave better performance than other algorithms. Performance of various comparable algorithms in terms of optimal operating time is given in table 4.

Table 3. Optimal setting of relays for 8 bus test system

Relay No.	TSM	Ip	Relay No.	TSM	Ip
1	0.1	0.5	8	0.1	0.5
2	0.1	0.5	9	0.1	1.5
3	0.1	0.5	10	0.1	0.5
4	0.1	0.5	11	0.1	0.5
5	0.1	0.8	12	0.1	0.5
6	0.1	0.5	13	0.1	0.5
7	0.1	0.5	14	0.1	0.5
Objective function value (J)				2.6245	

Table 4. Comparisons of optimal setting of relays with various algorithms

Algorithm	Objective function values (J)
Genetic Algorithm (GA)	2.7620 s
Particle Swarm Optimization (PSO)	3.6272 s
Enhanced Discrete Differential Evolution (EDDA)	2.6245 s

5 Conclusion

The details of the proposed algorithm and program implementation for directional overcurrent relay coordination have been discussed in this paper. The problem has been formulated as non-linear problem taking into account both the parameter i.e., TDS and Ip as discrete. Computer results for the 8 bus test system have been reported and compared with different algorithms. The obtained results show that the proposed algorithm (EDDEA) gave better optimal operating time of relay without miscoordination when compared to other algorithms.

References

[1] Ramaswami, R.: Transmission Protective Relay Coordination- A Computer-Aided-Engineering Approach for Subsystems and Full Systems, Ph.D. Dissertation, University of Washington Seattle (1986)

[2] Urdenta, A., Restrepo, H., Marques, S., Sanches, J.: Optimal Co-ordination of directional overcurrent relay timing using linear programming. IEEE Trans. Power Delivery 11, 122–129 (1996)

[3] Urdenta, A., Nadira, R., Prez, L.: Optimal Co-ordination of directional overcurrent relay in interconnected power system. IEEE Trans. Power Delivery 3, 903–911 (1988)

[4] Braga, A.S., Saraiva, J.T.: Co-ordination of directional overcurrent relays in meshed networks using simplex method. In: Proc. IEEE MELECON Conf., vol. 3, pp. 1535–1538 (1996)

[5] Abyaneh, H.A., Keyhani, R.: Optimal co-ordination of overcurrent relays in power system by dual simplex method. In: Proc. AUPEC Conf., Perth, Australia, vol. 3, pp. 440–445 (1995)

[6] Birla, D., Maheshwari, R.P., Gupta, H.O.: Time overcurrent relay coordination: a review. International Journal of Emerging Electric Power Systems 2(2), 1–13 (2005)

[7] Birla, D., Maheshwari, R.P., Gupta, H.O., Deep, K., Thakur, M.: Application of random search technique in overcurrent relay coordination. International Journal of Emerging Electric Power Systems 7(1), 1–14 (2006)

[8] So, C.W., Li, K.K., Lai, K.T., Fung, K.Y.: Application of genetic algorithm for overcurrent relay coordination. In: Proc. IEE Conference on Developments in Power System Protection, pp. 66–69 (1997)

[9] Zeineldin, H.H., El-Saadany, E.F., Salama, M.M.A.: Optimal coordination of overcurrent relays using a modified particle swarm optimization. Electrical Power System Research 76, 988–995 (2006)

[10] Qin, A.K., Huang, V.L., Suganthan, P.N.: Differential Evolution Algorithm with Strategy Adaptation for Global Numerical Optimization. IEEE Trans. Evolutionary Computation 13, 398–417 (2009)

[11] Bashir, M., Taghizadeh, M., Sadeh, J., Mashhadi, H.R.: A new hybrid particle swarm optimization for optimal coordination of overcurrent relay. In: International Conference on Power System Technology, pp. 1–6 (2010)

[12] Storn, R.: Differential evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces, ICSI, Tech. Rep. TR-95-012 (1995)

[13] Das, S., Suganthan, P.N.: Differential Evolution: A Survey of the State-of-the-Art. IEEE Trans. Evolutionary Computation 15(1), 4–31 (2010)

Dynamic Thinning of Antenna Array Using Differential Evolution Algorithm

Ratul Majumdar, Aveek Kumar Das, and Swagatam Das

Jadavpur University, Kolkata-700032, India
`{majumdar.ratul,aveek23}@gmail.com, swagatamdas19@yahoo.co.in`

Abstract. Dynamic thinning is a process by which the total number of active elements in an antenna array is reduced under real time conditions without causing major degradation in system's performances. This paper suggests a technique by which a thinned antenna array is designed by simple differential evolution(DE) algorithm overcoming the difficulties of large and rugged solution space and slow rate of convergence in case of large 2-D arrays.

Keywords: Antenna Array,Dynamic thinning, Differential Evolution, Zoning Technique.

1 Introduction

Thinning refers to strategic elimination of a subset of active elements within an antenna array, keeping the deviation of the resulting radiation pattern within limits. All the results present in this paper are based on study of Array factor of a set of isotropic radiators placed at a uniform spacing of half wavelength.

The array factor of a linear array of M identical antenna elements is given by,

$$F_M(\theta) = \sum_{i=1}^{M} R_i e^{jkd_i \cos(\theta)} \tag{1}$$

where R_i is the complex excitation coefficient of the i^{th} element located at $x = d_i$ along the linear array direction x and k is the wave number. Now to obtain a new linear array that has minimum number of elements which differs from, $F_M(\theta)$ by less than a prescribed tolerance of ε, with all other conditions remaining the same. So, we have to find a solution to,

$$\left\{ \min_{\{R_i, d_i\}_{i=1,\ldots,Q}} \left\| F_M(\theta) - \sum_{i=1}^{Q} R_i e^{jkd_i \cos(\theta)} \right\| \right\} \le \varepsilon \tag{2}$$

where R_i and d_i $(i = 1,\ldots,Q \le M)$ are the complex excitations and locations for Q antenna elements respectively, for all values of θ.

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 94–101, 2011.
© Springer-Verlag Berlin Heidelberg 2011

2 Objective Function

Construction of a good objective function is an important issue in any antenna array design problem.

In this paper the following objective function was used for the thinned antenna array synthesis:

$$\varepsilon(\theta) = [1+\omega_1 \left|\theta_0 - \theta_m^{(n)}(v)\right|^2].SLL_{\max}^{(n)}(v).[1+\omega_2 U(SLL_{\max}(v) - SLL_0)] \quad (3)$$

where (n) is the number of evolution generations, $\omega_i\,(i = 1,2)$ is the weighing factor for the different terms to the objective function. $SLL_{\max}^{(n)}(v)$ is the calculated maximum SLL and θ_o and θ_M are desired and calculated main beam directions respectively. In equation (3), U is a step function used for restriction of $SLL_{\max}^{(n)}(v)$ for prescribed SLL_0 such that $U = 1$ for $SLL_{\max}^{(n)}(v) > SLL_0$ and $U = 0$ for $SLL_{\max}^{(n)}(v) \le SLL_0$.This objective function is found to be ideal for using with simple differential evolution (DE) algorithm.

3 Overview of Differential Evolution

Differential evolution algorithm, proposed by Storn and Price [1] in 1995, is a simple but effective population-based stochastic search technique for solving global optimization problems. The algorithm is named differential evolution (DE) owing to a special kind of differential operator, which they invoked to create new off-spring from parent chromosomes instead of classical crossover or mutation.

Let $S \subset \Re^n$ be the **D** dimensional search space of the problem under construction. The D.E. evolves a population of **NP,** D-dimensional individual vectors, $\vec{X}_{i,G} = (x_{i,1}, x_{i,2}, \cdots, x_{i,D}) \in S, i = 1,2\ldots, NP$ from one generation to another. The primary features of D.E can be stated as follows [1, 2]:

3.1 Initialization

The 1st step of D.E. Algorithm is the **initialization of the population**. The population should ideally cover the entire search space between the prescribed upper and lower bounds $\vec{X}_{\max}$ and $\vec{X}_{\min}$ where $\vec{X}_{\max} = \{x_{1,\max}, x_{2,\max}, \ldots\ldots, x_{D,\max}\}$ and $\vec{X}_{\min} = \{x_{1,\min}, x_{2,\min}, \ldots\ldots. x_{D,\min}\}$. The j^{th} component of the i^{th} individual is initialized as follows:

$$x_{i,j} = x_{j,\min} + rand_i^j(0,1)\ (x_{j,\max} - x_{j,\min})\,;\ \text{j} \in [1,\text{D}] \quad (4)$$

Here $rand_i^j(0,1)$ is a uniformly distributed random number within [0, 1].

3.2 Mutation

For every **target vector,** $\vec{X}_{i,G}$, in any generation G, **a mutant vector** $\vec{V}_{i,G}$ is generated. The most common mutation policies generally used in DE are:

1. DE/rand/1/bin: $\vec{V}_{i,G} = \vec{X}_{r1,G} + F.(\vec{X}_{r1,G} - \vec{X}_{r3,G})$ (5)
2. DE/best/1/bin: $\vec{V}_{i,G} = \vec{X}_{best,G} + F \cdot (\vec{X}_{r1,G} - \vec{X}_{r2,G})$ (6)
3. DE/target-to-best/1/bin: $\vec{V}_{i,G} = \vec{X}_{i,G} + F.(\vec{X}_{best,G} - \vec{X}_{i,G}) + F_1.(\vec{X}_{r1,G} - \vec{X}_{r2,G})$ (7)

where r1, r2, and r3 are random and mutually exclusive integers generated in the range **[1, NP],** which should also be different from the trial vector's current index i. **F and F$_1$** are weight factors in [0, 2] for scaling differential vectors and $\vec{X}_{best,G}$ is the individual vector with best fitness value in the population at generation **G.**

3.3 Crossover

This operation involves **binary crossover** between the target vector $\vec{X}_{i,G}$ and the mutant vector $\vec{V}_{i,G}$ produced in the previous step which produces **the trial vector** $\vec{U}_{i,G} = (u_{i,1,G}, u_{i,2,G}, \ldots, u_{i,n,G})$. The crossover operation is done as follows.

$$u_{\mathrm{i,j,G}} = \begin{cases} v_{i,j,G} & if\ rand(0,1) \le CR\ or\ j = j_{rand} \\ x_{i,j,G} & otherwise \end{cases} \tag{8}$$

where ***CR*** is a user-specified crossover constant in the range **[0, 1)** and j_{rand} is a randomly chosen integer in the range **[1, NP]** to ensure that the trial vector $\vec{U}_{i,G}$ will differ from its corresponding target vector $\vec{X}_{i,G}$ by at least one parameter.

3.4 Selection

If the values of some parameters of a newly generated trial vector exceed the corresponding upper and lower bounds, we randomly and uniformly reinitialize it within the search range. The fitness value of each trial vector **f** ($\vec{U}_{i,G}$) is compared to that of its corresponding target vector **f** ($\vec{X}_{i,G}$) in the current population and the population for the **<u>G+1 generation</u>** is formed as follows: (for a minimization problem)

$$\vec{X}_{i,G+1} = \begin{cases} \vec{U}_{i,G} & if\ f(\vec{U}_{i,G}) \le f(\vec{X}_{i,G}) \\ \vec{X}_{i,G} & otherwise \end{cases} \tag{9}$$

where **f (.)** is the objective function.

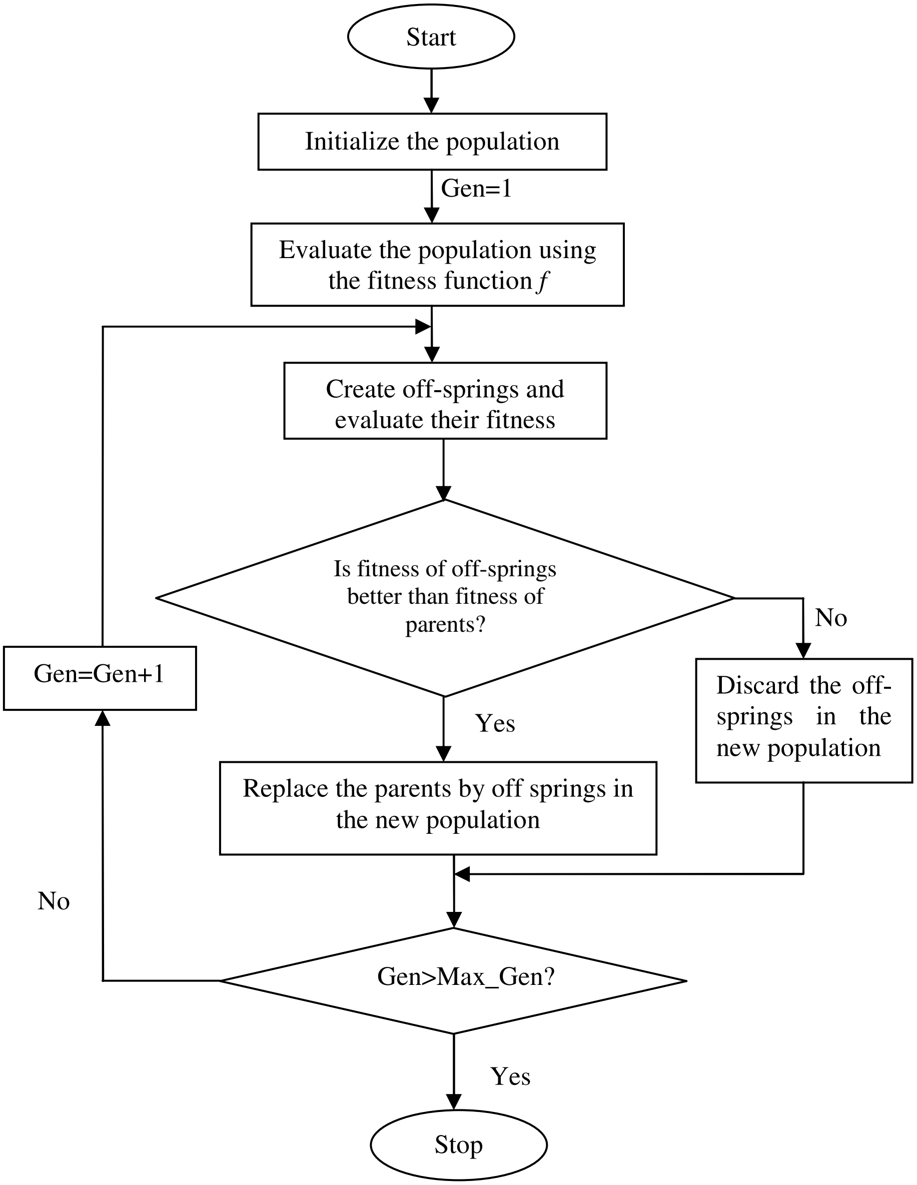

Start
Initialize the population
Gen=1
Evaluate the population using the fitness function *f*
Create off-springs and evaluate their fitness
Is fitness of off-springs better than fitness of parents?
No
Discard the off-springs in the new population
Yes
Replace the parents by off springs in the new population
Gen=Gen+1
No
Gen>Max_Gen?
Yes
Stop

Fig. 1. The Flowchart describes the Differential Evolution Operation

4 Application of Differential Evolution to Dynamic Array Thinning

It can be shown that the Differential Evolution (DE) algorithm is successful in designing a thinned array which can be accomplished by strategically removing a subset of active elements in the array to achieve the objective function. Now, this type of problem can be represented in a binary format by representing the active elements as value of ‘1’ and inactive elements are represented by value of ‘0’. So the problem reduces to finding the appropriate pattern of ‘1’s and ‘0’s in order to satisfy the constraints.

The first step of applying DE to the above is to initialize a set of random binary strings as the initial population where a ‘1’ represents an active element and a ‘0’ represents an inactive element in a particular string. Then the basic steps of DE involving mutation, recombination and selection is done iteratively on the population until end criteria is met. The Side Lobe Level (SLL) so obtained is compared with that of Simple Genetic Algorithm (SGA).It is seen from fig.2 that for a 100 element uniformly excited antenna array compared to the -20.5dB peak SLL of thinned antenna array by SGA the peak SLL by DE algorithm is -23dB.In both the cases a total of 20 array elements are found to be inactive.

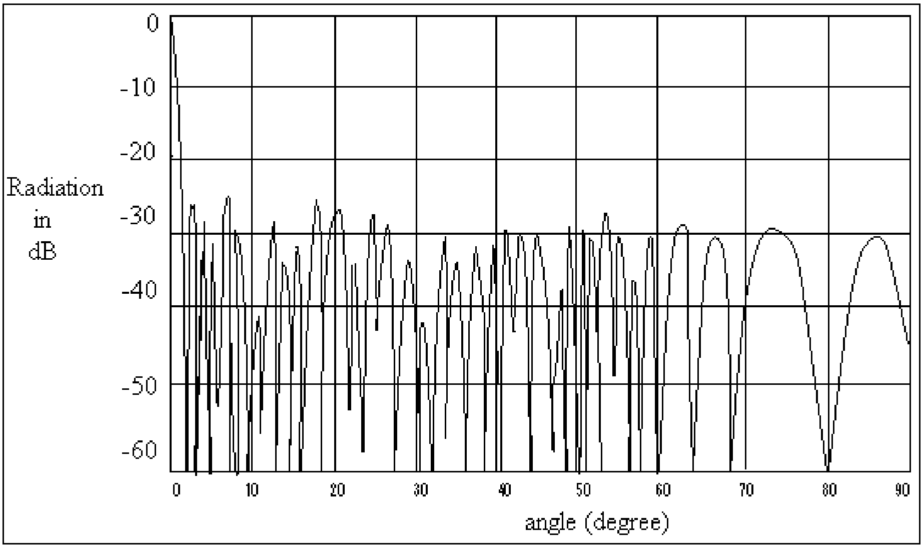


Fig. 2. Radiation pattern of a uniformly excited array after thinning using DE; Due to symmetry only half the pattern is shown

4.1 Dynamic Array Thinning

Dynamic array thinning is a process by which thinning pattern is varied on a real time basis to adapt to varying conditions. Although it is seen that performance of DE algorithm is better than SGA in solving thinned antenna arrays it still suffers the problem of computational complexity and finding an 'acceptable solution' for the real world scenario.

1. Computational Complexity: The evaluation of objective function usually involves long procedures. Also, for linear 2-D antenna arrays as the number of antenna elements increases the solution space within which a required solution can be found also increases. As a result exploring such a large solution space would not only require time but there is a chance of premature convergence.
2. Concept of Acceptable Solution: Generally an optimization problem would strive at getting at globally optimized solution within the set constraints. However, in a real-time scenario as in the case of dynamic thinning, a solution which gives a near-optimum value would be more useful input than the absolute minimum value. So, dynamic thinning should look for acceptable solution rather than the absolute minimum value.

The computational complexity and the problem of acceptable solution can be taken care of by the zoning technique [3] which is described below:

4.2 Zoning Technique

Zoning is the process by which the antenna array is partitioned into a number of convenient zones N_Z so that the solution space can be explored usefully. It is assumed that each zones consists of a minimum of two elements and $N_Z=1$ represents an undivided antenna array.

It is evident that the number of inactive elements in the central portion of the linear array is generally less than the other portions of the array. Such an assumption helps in better exploitation of the solution space. Thus, Zoning can help in enhancing convergence rate, since the proportion of the exploring space to total solution space reduces drastically in case of large arrays.

5 Parameter Values Used for the Experiment

The parameters for the Differential Evolution algorithm in this particular experiment performed are:

1. **Weight Factor: F and F1** are both taken as

$$F = F1 = F\_weight + rand(0,1)*F\_weight; \quad \mathbf{F\_weight = 0.5} \tag{10}$$

2. **Crossover Constant:** The value of the **Crossover constant (CR)** is taken as **0.7**
3. **Mutation Scheme of DE:** The mutation scheme in the DE part of the algorithm applied to the superior sub-population is the **"DE/target-to-best/1/bin"** scheme that follows the mutation according to equation (7)

6 Experimental Results

The simulation was carried on symmetrical arrays with isotropic radiators placed at a uniform spacing of half wavelength. Study was mainly carried out to see the effect of the number of zones on the speed of convergence. The Zoning details of 200 element linear antenna array are shown in table1.

The objective function was chosen to achieve -24dB SLL outside the sector of ± 0.01 radians. Maximum number of iterations taken is $N_{it} = 1000$ and the termination criterion is to achieve either the SLL or the maximum number of iterations whichever was earlier. The average of 50 runs of the said algorithm is shown below.

The convergence behavior of DE was compared with SGA [3] after observing

(i)Number of runs in which objective function was achieved within N_{it}.

(ii)Number of iterations required to achieve the desired SLL averaging over 50 runs.

(iii)SLL actually obtained while termination criteria is satisfied averaging over 50 runs.

Figure 3 shows the convergence trend of normalized SLL for the four zoning approaches used. For each curve, at every iteration, an average value of normalized SLL over 50 runs is calculated.

Table 1. Comparative Study of the SLL for DE and SGA algorithms

	Algorithm used	$N_z = 1$	$N_z = 2$	$N_z = 3$	$N_z = 4$
Zoning details (Number of elements in half array)		Zone1:100	Zone 1: 40 Zone 2: 60	Zone 1: 40 Zone 2: 30 Zone 3: 30	Zone 1: 40 Zone 2: 20 Zone 3: 20 Zone 4: 20
Number of runs in which desired SLL was achieved within N_{it}	DE	1	80	20	3
	SGA	1	90	40	3
Number of iterations to achieve desired SLL averaged over 50 runs	DE	457	346	358	330
	SGA	496	337	382	448
SLL (dB) obtained while termination, averaged over 50 runs	DE	-21.36	-23.15	-22.42	-23.68
	SGA	-20.35	-22.13	-22.03	-21.84

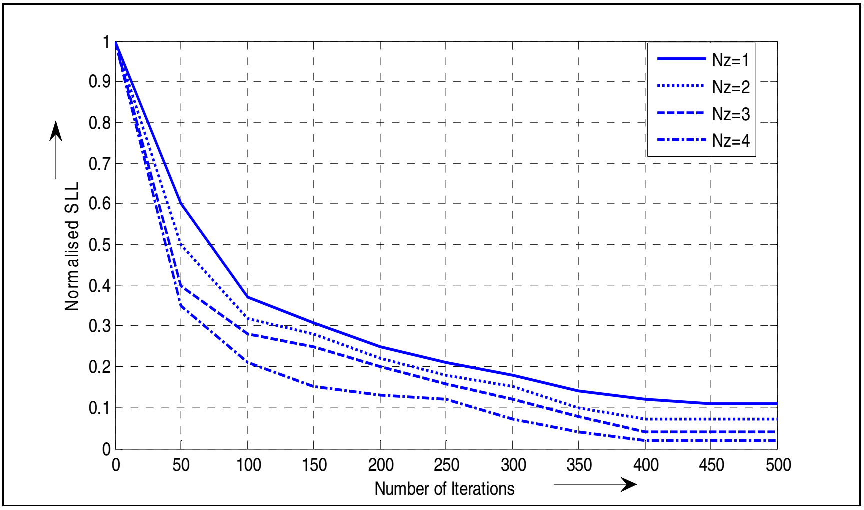


Fig. 3. Convergence curves for study of the number of zeroes

From the figure it is evident that best convergence is achieved for Nz=4.

7 Conclusion

Dynamic array thinning is a very important concept for real time thinning of antenna arrays in varying conditions. Simple differential evolution (DE) algorithm provides an easy and effective way to achieve this. However, for large 2-D arrays Zoning technique is employed for computational complexity and to solve the problem of 'acceptable solution'. The simulation results thus found were much better than simple genetic algorithm (SGA).

References

1. 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)
2. Das, S., Abraham, A., Konar, A.: Particle Swarm Optimization and Differential Evolution Algorithms: technical analysis, Applications and Hybridization Perspectives, `http://www.softcomputing.net/aciis.pdf`
3. Jain, R., Mani, G.S.: Dynamic Thinning Of Antenna Array Using Genetic Algorithm. Progress In Electromagnetics Research B 32, 1–20 (2011)
4. Bray, M.G., Werner, D.H., Boeringer, D.W., Machuga, D.W.: Optimization of thinned aperiodic linear phased arrays using genetic algorithms to reduce grating lobes during scanning. IEEE Trans. on Antennas and Propagation 50(12), 1732–1742 (2002)
5. Mahanti, G.K., Pathak, N., Mahanti, P.: Synthesis of Thinned Linear Antenna Arrays with Fixed Side lobe level using Real-Coded Genetic Algorithm. Progress In Electromagnetics Research, PIER 75, 319–328

A Quantized Invasive Weed Optimization Based Antenna Array Synthesis with Digital Phase Control

Ratul Majumdar, Ankur Ghosh, Souvik Raha, Koushik Laha, and Swagatam Das

Dept. of Electronics & Telecommunication Engg ,
Jadavpur University, Kolkata-700 032, India
{majumdar.ratul,ankurg708,rahasouvik37,lahakoushik1}@gmail.com,
swagatamdas19@yahoo.co.in

Abstract. The design of antenna arrays aims at minimizing side-lobe levels far as practicable. So a major concern of designer is to optimize the side lobes to increase directivity, gain, and efficiency. Invasive Weed Optimization (IWO) is a recently developed, ecologically inspired metaheuristic algorithm that has already found some major applications in electromagnetic research. In this article the synthesis of array antenna pattern by digital phase shifters is accomplished with a modified version of the IWO algorithm called QIWO (Quantized Invasive Weed Optimization. The proposed algorithm searches for optimal solution of the fitness function, which contains the SLL value and the interference suppression keeping the main beam unchanged. The results obtained from this algorithm are better than that of QPSO (Quantized Particle Swarm Optimization) and BPSO (Binary Particle Swarm Optimization). In this paper the array factor is expressed mathematically by a linear transform, which is similar to Discrete Cosine Transform (DCT). This proposed algorithm is also found to be efficient in computing for large arrays.

1 Introduction

In an array of identical elements, there are five controls that can be used to shape the overall pattern of the antenna array. These are (1) the geometrical configuration of overall array (linear, circular, rectangular, spherical etc.) (2) relative displacement between the elements, (3) excitation amplitude of individual elements (4) excitation phase of individual elements and (5) the relative pattern of individual elements. The theory related to this topic can be found in [1]-[3]. Among these the main factor of concern here is the excitation phase of individual array elements as these networks can minimize excitation errors and preserve coherence. In recent times, use of digital phase shifters has become very popular for beam scanning and interference suppression. Usually, the optimized phase solutions are calculated assuming the phase shifters to be variables that are continuous not discrete.. Thus, the actual calculated phases must be rounded to the nearest quantized value of digital phase shifters for simplification and also this is transferred into digital domain. Recently Mehrabian and Lucas proposed the Invasive Weed Optimization (IWO) [4], a metaheuristic algorithm, mimicking the ecological behavior of colonizing weeds. Here we use a modified version of IWO. For the particular problem we restrict the solution space

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 102–109, 2011.
© Springer-Verlag Berlin Heidelberg 2011

into finite discrete phase values. So the proposed Quantized IWO (QIWO) will search for an optimal solution within the available quantized values for digital phase shifters. Here the Side Lobe Levels (SLLs) are optimized for designing the antenna array [5] and the objective function includes a weighted sum of the mean beam width value, the sum of the nulls' depths at the interference signals directions, and the SLL value. To allow high speed solution of the objective or fitness functions, the array factor is expressed linearly, based on the Discrete Cosine Transform with pre-computed DCT matrix The proposed method provides significant performance improvement over Quantized Particle Swarm Optimization (QPSO) [6] and Binary Particle Swarm optimization (BPSO) [7, 8].

2 Digital Phase Antenna Array

Figure 1 shows a uniformly excited Centro-symmetric linear array of 2*N* isotropic equi-distant elements with inter element spacing *d*. Each element of the array is followed by a digital phase shifter φ. Thus the array factor with main beam steering is expressed as,

$$F(\theta)=\sum_{m=-N}^{-1} a(m).e^{j.m.d.k\left[\cos(\theta)-\cos(\theta_s)\right]}+$$
$$\sum_{n=1}^{N} a(n)\,e^{j.n.d.k\left[\cos(\theta)-\cos(\theta_s)\right]}, \tag{1}$$

where $a(n)$ is equal to $e^{j\varphi_n}$, *k* is the wave number, θ is the scanning angle, and θ_s is the main beam angle.

The first term corresponds to the array factor of the left hand half of the array, and the second term corresponds to the array factor of the right hand half of the array. By letting $m=-n$ the array factor can be written as:

$$F(\theta)=\sum_{n=N}^{1} a(n)e^{-jndk\left[\cos(\theta)-\cos(\theta_s)\right]}+\sum_{n=1}^{N} a(n)e^{jndk\left[\cos(\theta)-\cos(\theta_s)\right]}$$
$$=F_1(\pi-\theta)+F_2(\theta) \tag{2}$$

If the phase shifter values φ_n are taken in a symmetrical manner relative to the center of the array $a(n)=a(-n)$ then the Centro-symmetric array of size 2*N* is reduced to,

$$F(\theta)=2\sum_{n=1}^{N} a(n)\cos\left(n.d.k.\left(\cos(\theta)-\cos(\theta_s)\right)\right) \tag{3}$$

For one part of the array the arrival angle is θ while for the other part of the array the arrival angle is $\pi-\theta$. Now we discretize (3) with *M* points and the above equation can be written in matrix form by multiplication of two matrices having sampled values of a(n), & W(θ_i, n) with, $W(\theta_i, n) = \cos(ndk(\cos(\theta_i) - \cos(\theta_s)))$. Clearly the product of matrices containing the sampled values of $a(n)$ and W(θ_i, n) gives the same result as obtained by its continuous equivalent i.e. eqn. (3):

$$\begin{bmatrix} F(\theta_1) \\ \vdots \\ F(\theta_N) \end{bmatrix} = 2 \begin{bmatrix} W(\theta_1,1) & \cdots & W(\theta_1,N) \\ \vdots & \ddots & \vdots \\ W(\theta_M,1) & \cdots & W(\theta_M,N) \end{bmatrix} \begin{bmatrix} a(1) \\ \vdots \\ a(N) \end{bmatrix} \tag{4}$$

The above transformation is similar to performing discrete cosine transform.

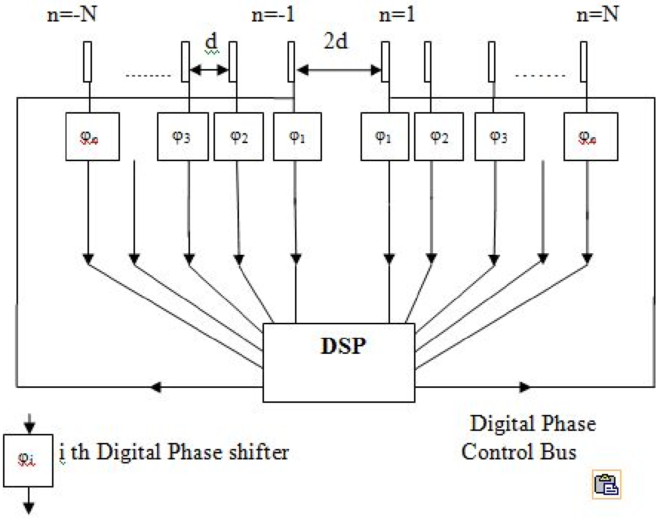


Fig. 1. Centro-symmetric antenna array using digital phase shifters

3 IWO and Its Proposed Modification

IWO is a population-based algorithm based on trial and error method that copies the colonizing behavior of weeds. Weed grows its population entirely or in a specified area without any control of any external factor in a very random manner. Initially a certain number of weeds are randomly spread over the entire search range. These weeds will eventually grow up and execute the following steps and the algorithm proceeds.

3.1 Classical IWO

There are four steps in the classical IWO algorithm as described below:

Initialization. A finite number of weeds are initialized randomly in the search space.

Reproduction. Each member of the population is allowed to produce seeds depending on its own, as well as the colony's lowest and highest fitness, such that, the number of seeds produced by a weed increases linearly from lowest possible seed for a weed with the worst fitness to the maximum number of seeds for a weed with the best fitness.

Spatial distribution. The generated seeds are randomly scattered over the *d*-dimensional search space by perturbing them with normally distributed random numbers with zero mean and a variable variance. However, the standard deviation (*sd*) of the random function is made to decrease over the iterations. If $sd_{\max}$ and $sd_{\min}$ are the maximum and minimum standard deviations and if pow is a real number, then the standard deviation for a particular iteration can be given as in eqn. (5):

$$sd_{ITER} = \left(\frac{iter_{\max} - iter}{iter_{\max}} \right)^{pow} \left(sd_{\max} - sd_{\min} \right) + sd_{\min}, \tag{5}$$

Where *iter* is the current iteration number and $iter_{\max}$ equals the maximum number of iterations allowed.

Competitive Exclusion. Some kind of competition between plants is needed for limiting maximum number of plants in a colony. Initially, the plants in a colony will reproduce until the number of plants reaches a maximum value $pop_{\max}$. From then on, only the fittest plants up to pop_max, among the existing ones and the reproduced ones, are taken in the colony and steps 2 to 4 are repeated until the maximum number of iterations has been reached, i.e. the colony size is fixed from thereon to $pop_{\max}$. This method is known as competitive exclusion and is also a selection procedure of IWO.

3.2 The Quantized IWO

The algorithm described above is based on continuous variables which are difficult to perceive in this context of digital phase shifters. This forced us to restrict the solution space into some quantized values of digital phase shifters consequently; QIWO will search for optimal solution within the available quantized values.

The algorithm starts by randomly initializing integer particle values. Each value or level *l* has the ability to change to a higher value *l*+1 or to a lower value *l*-1.Any value exceeding the maximum level *L*-1 is set to *L*-1, and any negative value is set to the lowest level 0.After bounding the positions and quantization is performed the fitness of each solution is obtained and the best solution is found by IWO.

In our optimization problem, a plant is the set of digital values of the phase shifters associated with each element of the half of the antenna array of size N whose position is given in N dimensional space by $X_q = \varphi_q = \lfloor \psi_{q1}, \psi_{q2}, \ldots\ldots, \psi_{qN} \rfloor$ where φ_q is the phase of an element of the antenna array. Each phase can have any value in the discrete space [0…15] since the total phase has been quantized in 16 divisions and therefore can be written with hexadecimal notation [0…F] which is encoded using 4 bits. Quantization means discrete amplitude representation of a physical quantity. .We contemplate the original physical quantity whose excursion is confined to a certain range ,the total range being divided into equal intervals .In the center of each of these intervals we locate quantization levels. At every instant of time, the approximated version of the physical quantity takes the value of the quantization level to which the recent value of the original physical quantity is closest.

A: minimum SLL optimization:

$$F1 = \min(AF(\theta)), \tag{6}$$

Where array factor is a function of θ

B: optimization with interference suppression:

$$F_2 = C_1 + C_2, \tag{7}$$

$$\text{Where } C_1 = K(SLL_O - SLL_d)^2 \tag{8}$$

$$\text{And } C_2 = \sum_{\theta=-90^o}^{\theta=+90^0} [W(\theta)\left|AF_0(\theta) - AF_d(\theta)\right|] \tag{9}$$

The first term of the cost function in (7) is used for the purpose of SLL suppression. SLL_0 and SLL_d in (8) are respectively the normalized side lobe level of actual pattern and the desired pattern. Coefficient *k* is the weight, controlling the optimization process. This second term of the cost function in (7) is for interference suppression i.e. to place nulls at specified directions, where, $AF_0(\theta)$ is the pattern obtained by using QIWO, $AF_d(\theta)$ is the desired pattern, and $W(\theta)$ is the controlling parameter for creating the null. Now we have applied the proposed QIWO algorithm on the final objective function $F_3(\theta)$ defined as:

$$F_3(\theta) = \mu_1 F_1(\theta) + \mu_2 F_2(\theta), \tag{10}$$

Where μ_1 and μ_2 are the weights signifying the importance of the individual terms.

4 Experiments and Results

The computing was achieved on a Pentium 4 computer running at 3.0 GHz equipped with 1.5 GB of RAM. Results are presented for arrays of 100 and 200 elements for a Centro-symmetric antenna. The only controlled parameter of the array is the element digital phase shifter which is 4 bit in all the following results.. In general, the values of the phases can have any discrete values between –Π and + Π .Running a large number of computer simulation examples we find that that a 4 bit phase shifter is a good choice with shift angles range set from 0° to 120° with 6.50° increment step. In term of computation time and based on the DCT implementation the search of optimum solutions among a space of 50,000 solutions is performed in only 15 seconds for an array size of 100 elements. This timing shows the efficiency of our implementation compared to QPSO algorithm.

4.1 Minimum SLL Optimization

First ,an array factor is designed with low side lobe level by optimizing the digital phase shifters. The optimization algorithm takes into consideration two linear equi-spaced isotropic antenna arrays with 100 array elements (*N*=50) and 200 array elements (*N*=100) respectively. Figures 4 and 5 show the array factor with SLL equal to 21 dB and 25 dB respectively while the main beam width remains equal to the uniformly excited antenna array. The values of SLL obtained is compared to the values obtained by QPSO in [8] and shown in Table I. An optimum set of discrete solutions of the digital phase shifters in provided in TABLE II. The array factor is taken every 0.5 degree.

Table 1. Comparison of QIWO with BPSO and QPSO

N	Values obtained by QPSO algorithm (dB)	Value obtained by BPSO algorithm (dB)	Value obtained by QIWO algorithm (dB)
50	18	17	**21**
100	21	20	**25**

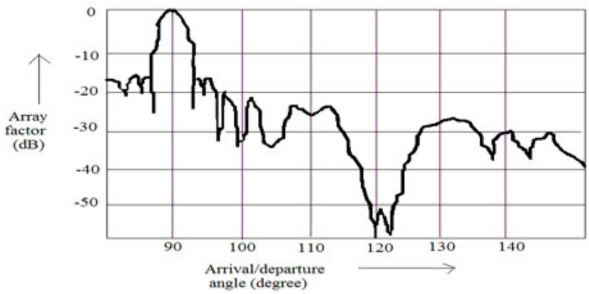


Fig. 2. The optimized array factor with minimum SLL only, d=λ/2, swarm size=100,SLL=21dB

Table 2. Different solutions given by their digital phase sequences and SLLs

FIGURE	DIGITAL PHASE SEQUENCE	SLL (dB)
2	000FA45888BDAC490EDC372A8 5375AC2D7E46091267549530025	21
3	145372EDCA437693BCEF465842 487635EDCABF333289BDE3D6327 452579ACDE	25
5	46676561345286235884685EDACFB AECFF888003442414238642346242 48800ADDEE4235	15
6	25000428289438ADEF42242CED472 4256356DEACBD24563556AED425 EFACEEBACE	17

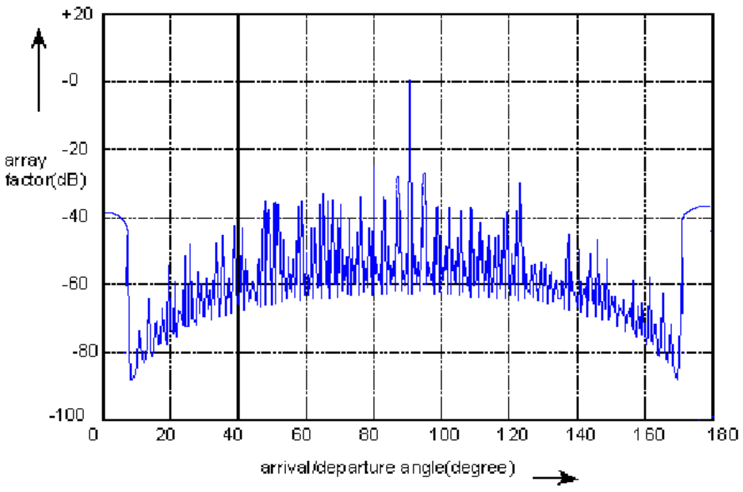


Fig. 3. The optimized array factor with minimum SLL only, d =λ/2, swarm size=200, SLL=25Db

To prove the effectiveness of QIWO over QPSO we plot the SLL value versus iteration value for N=50 averaging 20 trials in Fig. 4.The ash plot and the red plot shows the SLL values for QIWO and QPSO respectively.

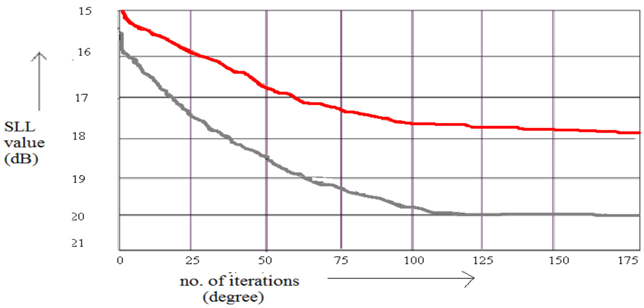


Fig. 4. SLL versus number of iterations using QPSO(red) and QIWO(black).the curves show the convergence by averaging 20 trials and taking N=50,swarm size=100.

Optimization with Interference Suppression. The digital phase shifters are used for narrowband and wideband interference suppression. The QIWO algorithm uses a swarm size of 100 and the fitness function given in (2).Fig. 5 shows two array factors with 15 dB SLL with one suppressed sectors of 1° starting from 120° compared to 12.4 dB SLL obtained from QPSO respectively. Also for the third fitness function we find that QIWO produces the best result for μ_1=0.6 & μ_2=0.4.

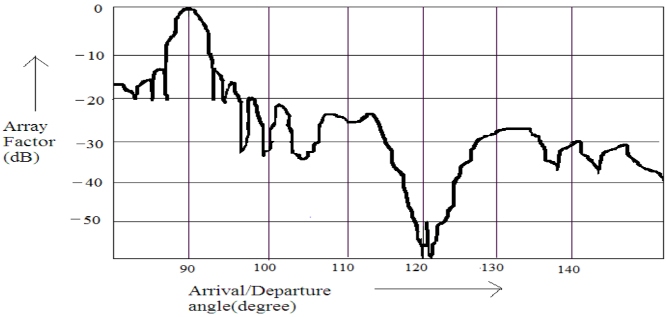


Fig. 5. The optimized array pattern with one suppressed sector (120º-121º),d= λ/2,N=50

5 Conclusion

The digital phase shifters are controlled for interference suppression while minimizing SLL by quantized invasive weed optimization (QIWO).The algorithm proposed here is more effective than QPSO because IWO in many applications gives better results than PSO and is faster than other algorithms. The particles with better fitness are allowed to reproduce and hence they find the optimum solution for the multimodal function more efficiently and quickly.

References

1. Baird, D., Rassweiler, G.: Adaptive sidelobe nulling using digitally controlled phase shifters. IEEE Trans. Antennas Propag. 24(5), 638–649 (1976)
2. Steyskal, H., Shore, R.A., Haupt, R.L.: Methods for null control and their effects on radiation pattern. IEEE Trans. Antennas Propag. 34, 404–409 (1986)
3. Haupt, R.L.: Phase-only adaptive nulling with a genetic algorithm. IEEE Trans. Antennas Propag. 45(6), 1009–1015 (1997)
4. Mehrabian, A.R., Lucas, C.: A novel numerical optimization algorithm inspired from weed colonization. Ecological Informatics 1, 355–366 (2006)
5. Oliveri, G., Poli, L.: Synthesis of monopulse sub-arrayed linear and planar array antennas with optimized sidelobes. Progress In Electromagnetics Research, PIER 99, 109–129 (2009)
6. Ismail, T.H., Hamici, Z.M.: Array pattern sunthesis using digital phase control by quantized particle swarm optimization. IEEE Transactions on Antennas and Propagation 58(6), 2142–2145 (2010)
7. Jin, N., Rahmat-Samii, Y.: Advances in particle swarm optimization for antenna designs: Real-number, binary, single-objective and multiobjective implementations. IEEE Trans. Antennas Propag. 55(3), 556–567 (2007)
8. Kennedy, J., Eberhart, R.C.: A discrete binary version of the particle swarm algorithm. In: IEEE Int. Conf. Man Syst., vol. 5, pp. 4104–4108 (October 1997)

Optimal Power Flow for Indian 75 Bus System Using Differential Evolution

Aveek Kumar Das[1], Ratul Majumdar[1],
Bijaya Ketan Panigrahi[2], and S. Surender Reddy[2]

[1] Department of E.T.C.E, Jadavpur University, Kolkata
[2] Department of E.E, Indian Institute of Technology, Delhi
{aveek23,majumdar.ratul,bijayaketan.panigrahi,
salkuti.surenderreddy}@gmail.com

1 Introduction

The optimal power flow (OPF) has been commonly used as an efficient tool in the power system planning and operation for many years. OPF is an important tool in modern Energy Management System (EMS). It plays an important role in maintaining the economy of the power system. The problem of OPF subjected to set of equality and inequality constraints was originally formulated in 1962 by Carpentier [1]. The OPF problem is a nonlinear, non-convex, large scale, static optimization problem with both continuous (generator voltage magnitudes and active powers) and discrete (transformer taps and switchable shunt devices) control variables. Even in the absence of discrete control variables, the OPF problem [2] is non convex due to the existence of the nonlinear (AC) power flow equality constraints. The presence of discrete control variables, such as switchable shunt devices, transformer tap positions further complicates the problem solution.

Evolutionary Algorithms (EAs) such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Differential Evolution (DE) etc. have become the method of choice for optimization problems that are too complex to be solved using deterministic techniques such as linear programming or gradient (Jacobian) methods.

EAs require little knowledge about the problem being solved, and they are easy to implement, robust and inherently parallel. Because of their universality, ease of implementation, and fitness for parallel computing, EAs often take less time [3] to find the optimal solution than gradient methods.

OPF can be used periodically to determine the optimal settings of the control variables to minimize the generation cost, minimization of the transmission system losses. Genetic Algorithm (GA), Enhanced GA, Improved GA has been successfully applied for solution of OPF problem [4]-[6]. In [7], PSO was used to solve the OPF problem. Here, we propose the Differential Evolution algorithm, and a Hybridized Algorithm with D.E and I.W.O (with Population Exchange) which can handle both continuous and discrete control variables. The effectiveness of the differential evolution algorithm is tested on practical Indian 75 bus system data.

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 110–118, 2011.
© Springer-Verlag Berlin Heidelberg 2011

2 Optimal Power Flow (OPF) Problem Formulation

The OPF problem is to optimize the steady state performance of the power system in terms of the objective function while satisfying several equality and inequality constraints. OPF is a highly non-linear, non-convex, large scale static optimization problem due to large number of variables and limit constraints. It minimizes a certain objective function subjected to a set of equality and inequality constraints.

Mathematically, the problem was defined as

$$\begin{aligned} &\text{Minimize} && f(x, u, p) \\ &\text{Subjected to} && g(x, u, p) = 0 \\ & && h(x, u) \le 0 \end{aligned}$$

Where $x^T = [\delta, V]$

$$u^T = [V_{G_1} \ldots V_{G_{NG}}, P_{G_2} \ldots P_{G_{NG}}, T_1 \ldots T_{NT}, Q_{C_1} \ldots Q_{C_{NC}}]$$

The minimization function ‘f’ (objective function) can take different forms.

2.1 Objective Functions

Fuel Cost (FC) Minimization

For active power optimization (economic load dispatch of the thermal units) fuel cost is considered as the objective function. The minimization function f can be obtained as sum of the fuel costs of all the generating units.

$$f = \sum_{i=1}^{NG} F_i(P_{Gi}) \tag{1}$$

$$F_i(P_{Gi}) = a_i + b_i P_{Gi} + c_i P_{Gi}^2 \tag{2}$$

Where a_i, b_i, c_i are cost coefficients of unit-i, P_{Gi} is real power generation of unit-i.

Real power losses (PL)

For reactive power optimization, system transmission loss minimization is considered as the objective function. Transmission power loss in each branch is calculated from the load flow solution. The converged load flow solution gives the bus voltage magnitudes and phase angles. Using these, active power flow through the lines can be evaluated. Net system power loss is the sum of power loss in each line.

$$f = \sum_{i=1}^{Nl} Loss_i \tag{3}$$

Where Nl is the number of transmission lines in a power system.

2.2 OPF Problem Constraints [8]

Equality constraints: These are load flow (nodal power balance) equations.

Inequality Constraints: These constraints represent system Operating limits.

Generator Constraints: Generator Voltage magnitudes V_G, Generator active power P_G and reactive power Q_G are restricted by their lower and upper limits.

Transformer Constraints: Transformer taps have minimum and maximum setting limits.

Switchable VAR sources: The switchable VAR sources have minimum and maximum restrictions on reactive power of Switchable VAR sources.

Security constraints: These include the limits on the load bus voltage magnitudes and line flow limits.

3 Overview of Differential Evolution

Differential evolution algorithm, proposed by Storn and Price [9] in 1995, is a simple but effective population-based stochastic search technique for solving global optimization problems. The algorithm is named differential evolution (DE) owing to a special kind of differential operator, which they invoked to create new off-spring from parent chromosomes instead of classical crossover or mutation. The main differences between Genetic Algorithm and DE algorithm are the selection process and the mutation scheme that makes DE self-adaptive.

Let $S \subset \Re^n$ be the **D** dimensional search space of the problem under construction. The D.E. evolves a population of **NP,** D-dimensional individual vectors, $\vec{X}_{i,G}$ = $\left(x_{i,1}, x_{i,2}, \cdots, x_{i,D}\right) \in S, i = 1,2\ldots, NP$ from one generation to another. The primary features of D.E can be stated as follows [9, 10]:

3.1 Initialization

The 1st step of D.E. Algorithm is the **initialization of the population**. The population should ideally cover the entire search space upper and lower bounds $\vec{X}_{\max}$ and $\vec{X}_{\min}$: $\vec{X}_{\max} = \{x_{1,\max}, x_{2,\max}, \ldots\ldots, x_{D,\max}\}$ and $\vec{X}_{\min} = \{x_{1,\min}, x_{2,\min}, \ldots\ldots x_{D,\min}\}$. The j^{th} component of the i^{th} individual is initialized as follows:

$$x_{i,j} = x_{j,\min} + rand_i^j(0,1)\ (x_{j,\max} - x_{j,\min})\ ; \mathrm{j} \in [1,\mathrm{D}] \tag{4}$$

Here $rand_i^j(0,1)$ is a uniformly distributed random number within [0, 1].

3.2 Mutation

For every **target vector,** $\vec{X}_{i,G}$, in any generation G, **a mutant vector** $\vec{V}_{i,G}$ is generated. The most common mutation policies generally used in DE are:

1. DE/rand/1/bin: $\vec{V}_{i,G} = \vec{X}_{r1,G} + F.(\vec{X}_{r1,G} - \vec{X}_{r3,G})$ (5)

2. DE/best/1/bin: $\vec{V}_{i,G} = \vec{X}_{best,G} + F \cdot (\vec{X}_{r1,G} - \vec{X}_{r2,G})$ (6)

3. DE/target-to-best/1/bin: $\vec{V}_{i,G} = \vec{X}_{i,G} + F.(\vec{X}_{best,G} - \vec{X}_{i,G}) + F1.(\vec{X}_{r1,G} - \vec{X}_{r2,G})$ (7)

Where r1, r2, and r3 are random and mutually exclusive integers generated in the range **[1, NP],** which should also be different from the trial vector's current index i. **F and F₁** are weight factors in [0, 2] for scaling differential vectors and $\vec{X}_{best,G}$ is the individual vector with best fitness value in the population at generation **G.**

3.3 Crossover

This operation involves **binary crossover** between the target vector $\vec{X}_{i,G}$ and the mutant vector $\vec{V}_{i,G}$ produced in the previous step which produces **the trial vector** $\vec{U}_{i,G} = (u_{i,1,G}, u_{i,2,G}, \ldots, u_{i,n,G})$. The crossover operation is done as follows.

$$u_{\mathrm{i,j,G}} = \begin{cases} v_{i,j,G} & if\ rand(0,1) \le CR\ or\ j = j_{rand} \\ x_{i,j,G} & otherwise \end{cases} \tag{8}$$

where ***CR*** is a user-specified crossover constant in the range **[0, 1)** and j_{rand} is a randomly chosen integer in the range **[1, NP]** to ensure that the trial vector $\vec{U}_{i,G}$ will differ from its corresponding target vector $\vec{X}_{i,G}$ by at least one parameter.

3.4 Selection

The fitness value of each trial vector **f** $(\vec{U}_{i,G})$ is compared to that of its corresponding target vector **f** $(\vec{X}_{i,G})$ in the current population and the population for the **G+1 generation** is formed as follows: (for a minimization problem)

$$\vec{X}_{i,G+1} = \begin{cases} \vec{U}_{i,G} & if\ f(\vec{U}_{i,G}) \le f(\vec{X}_{i,G}) \\ \vec{X}_{i,G} & otherwise \end{cases} \tag{9}$$

Where **f (.)** is the objective function.

4 Hybridized D.E and I.W.O with Population Exchange

The Invasive Weed Optimization (IWO) and DE algorithms are applied simultaneously on two sub-populations and population exchange is incorporated to refine the quality of the population after every generation. **Invasive Weed Optimization** [11] is a population-based algorithm based on trial and error method that copies the colonizing behavior of weeds [12].The main concept behind this population exchange is taken from the basic concept of any sports team, that, if the less-performing members of a team are replaced by the better-performing members, the team performs much better on an average. Thus the concept of population exchange rises.

Suppose the 2 sub-populations are updated every generation according to IWO and DE respectively. Now, after every generation(G), the best (i.e. with best fitness) "m" percent members of the 1st sub-population, after that iteration(G), is exchanged with the worst(i.e. with worst fitness) "m" percent members of the 2nd sub-population, thereby refining the 2nd population and making it more effective for the next generation(G+1). The same exchange can be done to the 1st sub-population to make the algorithm perform in a better way as a whole. The value of "m" is taken as an input and made equal to 25.

- 1st/ 2nd Sub-population for (G+1)th generation= m% of the best members of the 2nd/1st sub-population which replaces the worst m% + (100-m)% of the remaining members from the (G)th generation. (10)

5 Parameter Values Used for the Experiment

The parameters for in this particular experiment performed are:

1. **Population Size:** The **total size of the population (NP)** is taken to be **50**.
2. **Weight Factor : F and F1** are both taken as

   F= F1= F_weight + rand (0, 1)*F_weight; **F_weight =0.5** (11)

3. **Crossover Constant :**_The value of the **Crossover constant (CR)** is taken as **0.7.**
4. **Mutation Scheme of DE:** The mutation scheme in the DE part of the algorithm applied to the superior sub-population is the **"DE/target-to-best/1/bin"** scheme that follows the mutation according to equation (7).

5. **Limiting Values of Standard Deviation for IWO:** The values of sd_{min} and sd_{max} are 0.001 and 5 respectively.
6. **The Modulation Index for IWO:** The value of modulation index(pow) is taken as 4.5 (a sufficiently high value so as to magnify the search space for formation of new plants in IWO).

6 Results of the Algorithm Implementation

The Indian 75 Bus, 97 branch system consists of **fifteen generators, twenty-four transformers, ninety-seven lines and twelve shunt reactors**. In the optimization of OPF of the system there are **sixty four (64) control variables** of the system: 15 generator bus voltage magnitudes, 14 generator active powers (excluding the slack generator), 24 tap transformer settings and 12 shunt reactors. . The results given are the best solutions (minimum cost value) over 10 runs. Table 3 shows the control variables values and Table 4 shows the comparative fuel cost values between Hybridized D.E-I.WO, D.E, P.S.O and I.W.O. Fig. 1 shows the convergence characteristics of the Differential Evolution (DE) algorithm.

Table 1. Generator Cost Coefficients for Indian 75 bus system

Generator No.	Real Power Output Limit (MW)		Fuel Cost Coefficients		
	Maximum	Minimum	a	b	c
1	15	1	0.008	0.814	0
2	3	1	0.0014	1.3804	0
3	2	0.4	0.0016	1.5662	0
4	1.7	0.4	0.0016	1.6069	0
5	2.4	0.02	0.0016	1.5662	0
6	1.2	0.01	0.0018	1.7442	0
7	1	0.01	0.0018	1.7755	0
8	1	0.2	0.0018	1.7422	0
9	5.7	0.6	0.0012	1.1792	0
10	2.5	0.3	0.0017	1.6947	0
11	2.0	0.4	0.0016	1.6208	0
12	18	0.8	0.0004	1.4091	0
13	9	0.5	0.0007	0.6770	0
14	1.5	0.1	0.0015	1.4910	0
15	4.54	0.2	0.0010	1.0025	0

Table 2. Limits of Variables for 75 Bus System

Sl .No.	Description	Upper Limit	Lower Limit
1.	Generator Voltage	1.1	0.95
2.	Load Voltage	1.05	0.9
3.	Transformer Tap	0.9	1.1
4.	Shunt Reactors	0.0	0.05

Table 3. Control Variable Values corresponding to the best result

Generator Voltages		Generator Powers			Tap Transformers		Shunt Reactors	
Generator No.	Value (Volt)	Generator No.	Real Power (MW)	Reactive Power (MVAR)	Tap No.	Value	Shunt No.	Value
1	1.0300	2	100.000	89.400	1	0.9000	1	0.982
2	0.9759	3	148.288	235.482	2	1.0250	2	1.078
3	1.0139	4	71.0260	119.312	3	1.0875	3	0.955
4	1.0536	5	43.8954	73.618	4	0.9000	4	1.072
5	1.0627	6	7.07132	11.155	5	1.0875	5	0.960
6	1.0696	7	26.2905	47.048	6	1.0375	6	0.950
7	0.9874	8	25.9262	47.242	7	1.0625	7	1.008
8	0.9516	9	61.7397	114.424	8	1.0375	8	1.046
9	1.0937	10	60.0000	75.072	9	1.0750	9	1.100
10	1.0691	11	40.1591	70.799	10	1.0000	10	0.984
11	1.0973	12	40.0000	67.392	11	1.0000	11	1.070
12	0.9919	13	113.202	51.437	12	1.0125	12	1.080
13	1.0213	14	50.0000	35.600	13	1.1000		
14	1.0703	15	24.2167	36.987	14	0.9125		
15	0.9500				15	1.0250		
					16	0.9375		
					17	1.0500		
					18	1.0250		
					19	1.0875		
					20	1.0375		
					21	1.0625		
					22	1.0250		
					23	1.1000		
					24	1.0500		

The Losses of the system which was calculated by the 1st generator (slack generator) is found to be equal to 6.182217 MW.

Table 4. Comparison of the Best, Worst, Average and S.D of cost over 10 runs (Between hybridized D.E-I.W.O, D.E, P.S.O and I.W.O)

Cost($/hr)	Algorithm			
	Hybrid D.E.-I.W.O	D.E.	P.S.O	I.W.O
Best	1042.97812	1087.54028	1261.8568	1213.8650
Average	1071.22352	1124.67659	1306.6439	1265.9812
Worst	1117.89166	1209.0989	1344.0017	1296.0948
Standard Devn.	31.4594	34.39535	47.0876	42.8861

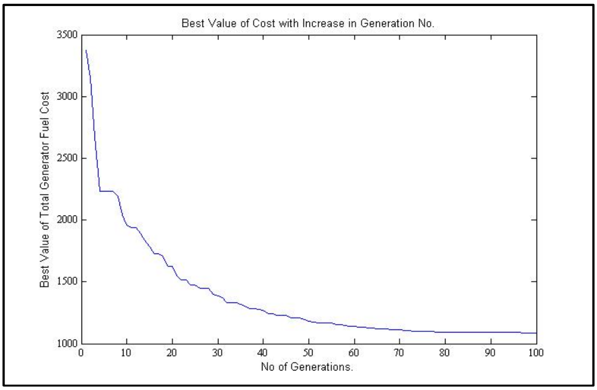


Fig. 1. Best Value of Cost vs No of Generations (best run)

7 Conclusions

This paper presents a DE solution to the optimal power flow problem and is applied to an Indian 75-bus power system. The main advantage of DE over other modern heuristics is modeling flexibility, sure and fast convergence, less computational time than other heuristic methods. And it can be easily coded to work on parallel computers.

The Future work in this area consists of applying other variants of D.E to the bus system and comparing the values obtained to the values obtained in this experiment.

References

1. Wood Allen, J., Bruce, F.: Power Generation Operation and Control. John Wiley and Sons, Inc., New York (1984)
2. Dommel, H.W., Tinney, W.F.: Optimal power flow solutions. IEEE Trans. Power App. Syst. 87(10), 1866–1876 (1968)
3. Sbalzarini, I.F., Muller, S., Koumoutsakos, P.: Multiobjective optimization using evolutionary algorithms, Center for Turbulence Research, Proceedings of the summer program (2000)
4. Osman, M.S., Abo-Sinna, M.A., Mousa, A.A.: A solution to the optimal power flow using genetic algorithm. Appl. Math. Comput. (2003)

5. Bakirtzis, A.G., Biskas, P.N., Zoumas, C.E., Petridis, V.: Optimal Power Flow by Enhanced Genetic Algorithm. IEEE Trans. on Power Syst. 17(2), 229–236 (2002)
6. Lai, L.L., Ma, J.T., Yokoyama, R., Zhao, M.: Improved genetic algorithms for optimal power flow under both normal and contingent operation states. Electrical Power & Energy Systems 19(5), 287–292 (1997)
7. Abido, M.A.: Optimal Power Flow using Particle Swarm Optimization. Electrical Power and Energy Systems 24(7), 563–571 (2002)
8. Sailaja Kumari, M., Sydulu, M.: An Improved Evolutionary Computation Technique for Optimal Power Flow Solution. International Journal of Innovations in Energy Systems and Power 3(1), 32–45 (2008)
9. 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)
10. Das, S., Suganthan, P.N.: Differential Evolution: A Survey of the State-of-the-art. IEEE Trans. on Evolutionary Computation 15(1), 4–31 (2011)
11. Mehrabian, A.R., Lucas, C.: A novel numerical optimization algorithm inspired from weed Colonization. Ecological Informatics, 355–366 (2006)
12. Dekker, J.: Course Works of Agronomy 517: A course on the biology and evolutionary ecology of invasive weeds (2005)

A Modified Differential Evolution Algorithm Applied to Challenging Benchmark Problems of Dynamic Optimization

Ankush Mandal, Aveek Kumar Das, and Prithwijit Mukherjee

Jadavpur University, ETCE department, Kolkata-700032, India
{ankushmandal19,aveek23}@gmail.com, mukh.prith@yahoo.com

Abstract. Many real-world optimization problems are dynamic in nature. In order to deal with these Dynamic Optimization Problems (DOPs), an optimization algorithm must be able to continuously locate the optima in the constantly changing environment. In this paper, we propose a multi-population based differential evolution (DE) algorithm to address DOPs. This algorithm, denoted by pDEBQ, uses Brownian & adaptive Quantum individuals in addition to DE individuals to increase the diversity & exploration ability. A neighborhood based new mutation strategy is incorporated to control the perturbation & there by to prevent the algorithm from converging too quickly. Furthermore, an exclusion rule is used to spread the subpopulations over a larger portion of the search space as this enhances the optima tracking ability of the algorithm. Performance of pDEBQ algorithm has been evaluated over a suite of benchmarks used in Competition on Evolutionary Computation in Dynamic and Uncertain Environments, CEC'09.

Keywords: Differential Evolution, Dynamic Optimization, Neighborhood based Mutation Strategy.

1 Introduction

The Differential Evolution (DE) [1-3] has become one of the most powerful tools in evolutionary algorithms (EAs) in recent years. However, unlike the traditional EAs, the DE-variants perturb the current-generation population members with the scaled differences of randomly selected and distinct population members. Therefore, no separate probability distribution has to be used, which makes the scheme self-organizing in this respect.

Several problems that we face in real world are dynamic in nature. For these Dynamic Optimization problems (DOPs), the function landscape changes over time. Practical examples of such situations are price fluctuations, financial variations etc. However, for dynamic environment, convergence problem is a significant limitation in case of EAs; if it becomes greatly converged, it will be unable to respond effectively to dynamic changes. So, in case of DOPs our main challenge is to maintain the diversity and at the same time to produce high quality solutions by tracking the moving optima.

In this paper, a multi-population based partial DE variant with Brownian & Quantum individuals, denoted by pDEBQ, is proposed to address DOPs. The main change that we

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 119–126, 2011.
© Springer-Verlag Berlin Heidelberg 2011

introduce in pDEBQ algorithm is the partial DE method. In each subpopulation, one individual is adaptive Quantum individual, analogous to particles in Quantum mechanics & one individual is Brownian individual whose trajectory is similar to the trajectory of any particle in Brownian motion; other individuals in the subpopulation evolve according to DE with a neighborhood based double mutation strategy. The double mutation strategy is developed to prevent the algorithm from converging quickly. In order to produce high accuracy in locating optima within the continuous search space, we have also introduced a control parameter to control the diversity.

However, in case of EAs, stagnation is another significant problem as it reduces adaptability & exploration ability. So, we incorporated an aging mechanism to prevent the algorithm from stagnation. Also an exclusion scheme is used so that the subpopulations become evenly distributed over the entire search space. This increases the exploration efficiency & optima tracking ability of the algorithm.

We tested the performance of our algorithm on a suite of DOPs generated by the Generalized Dynamic Benchmark Generator (GDBG) [4] which was used in Special Session & Competition on “Evolutionary Computation in Dynamic and Uncertain Environments”, CEC 2009.

2 Overview of Differential Evolution

Differential Evolution was first proposed by Storn & Price [1]. The main steps of DE algorithm are: 1) Initialization of the population – the search process starts with a initial population of target vectors, 2) Mutation – a mutant vector is created corresponding to each target vector following some mutation scheme, 3) Crossover operation - in this operation the mutant vector exchanges its component with the target vector to form a trial vector, 4) Selection operation – the trial vector is selected for the next generation if it shows a better fitness value than the target vector. These steps (except the Initialization step) are repeated generation after generation until a stopping criterion is fulfilled. A detailed survey on DE is given in [3].

3 The pDEBQ Algorithm

3.1 Partial DE Scheme

In order to ensure the population diversity, we introduced adaptive Quantum and Brownian individuals in addition to DE individuals. These Quantum and Brownian individuals do not follow the same rule as DE individuals to generate the next generation individuals. Hence, the population diversity is maintained to a greater extent.

Actually, within a subpopulation, two individuals are randomly chosen at each generation; we apply Quantum individual generation process to one of them and Brownian individual generation process to the other. If one of the chosen individuals is the best individual in the subpopulation then we randomly choose another individual and replace it with the best individual.

1) Adaptive Quantum Individuals: In quantum mechanics, due to the uncertainty in position measurement, position of a particle is probabilistically defined. Here we use this idea to generate individuals within a specific region around the local best position.

The steps for stochastically generating an individual whose position is inside a sphere of radius *R* and centered on the local best position $\vec{Lb}$ are as follows:

i) Generate a radius randomly from a uniform distribution within the range (0, R). $r = U(0, R)$

ii) Generate a vector whose each component is randomly taken from a normal distribution having mean=0 and variance=1. $\vec{X} = \{x_1, x_2, \ldots\ldots, x_D\}$; $x_d \in$ N (0, 1); *where* $1 \le d \le$ D

iii) Compute the distance of the vector from the origin. $Dist = \sqrt{\sum_{i=1}^{D} x_i^2}$

iv) New quantum individual's position will be:

$$\vec{Lb} + (r / Dist) * \vec{X} \tag{1}$$

The probability of generating a point near the local best point increases with dimensionality. In pDEBQ, the quantum individuals are adaptive because the radius within which the individuals are to be generated is self adaptive in nature, i.e. the radius is varied automatically according to the condition.

2) Brownian Individuals: Brownian motion is the mathematical model used to describe the random movement of particles suspended in a fluid. We used a very simple model to incorporate the Brownian motion into the positions of individuals. Actually, new individuals are generated within a Gaussian hyper parallelepiped. If the local best position is $\vec{Lb}$ then the new Brownian individual's position will be:

$$\vec{Lb} + \vec{N}\ (0, \sigma) \tag{2}$$

where σ is the standard deviation of the normal distribution from which the perturbation is randomly sampled.

3) DE Individuals: These individuals follow the DE method for generating new individuals. Here we have developed a new mutation scheme to produce mutant vectors. We have discussed this mutation scheme in the following section in details.

3.2 Double Mutation Strategy

In the case of DOPs, the optima positions are changed dynamically. If the population becomes greatly converged to a global optima then it will lose its ability to find the global optima again when the position of the global optima changes. So, here the idea is to control the perturbation to slow down the searching process. For this, we have developed a double mutation scheme. In this scheme, we first generate a mutant vector according to a neighborhood based mutation scheme and then we add this mutant vector to the local best vector with a weight factor having maximum value of 0.1. As we did the mutation process twice for each DE individuals, we call it double mutation strategy.

1) Neighborhood based Mutation Scheme: It's a modified *DE/current-to-best/2/bin* scheme. In order to overcome the limitations of fast but less reliable convergence performance of *DE/current-to-best/2/bin* scheme, we have made some changes in the process of generating two differential vectors. For the 1st differential vector, in our modified scheme we used the difference between nearest memory individual and the current individual where the memory archive is the collection of best individuals from the previous subpopulations. This change is done to control the convergence of the population towards global optima and encourage the subpopulations to explore the vicinity of local best position. For the second differential vector, we took the difference between the best individual in the neighborhood and the worst individual in the neighborhood with respect to the current individual. This second modification is done to guide the mutant vector to explore the neighborhood of the current individual within the subpopulation.

During the process of generating the mutant vector, the scaling factor is independently generated for each dimension of each differential vector and its value also depends on the magnitude of the differential vector along the corresponding dimension. Actually, for each dimension of each differential vector, the scaling factor is randomly generated from a uniform distribution within a range and this range is varied inversely with the magnitude of the differential vector along the corresponding dimension. This adaptation of the scaling factor is done to amplify the differential vector properly.

The first mutation according to neighborhood based mutation scheme can be expressed as:

$$\vec{V}^{j}_{mut,G} = \vec{X}^{j}_{i,G} + F^{j}_{m} \cdot (\vec{X}^{j}_{m,G} - \vec{X}^{j}_{i,G}) + F^{j}_{bw} \cdot (\vec{X}^{j}_{best,G} - \vec{X}^{j}_{worst,G}) \text{,where } j \in \{1,2,.......,D\} \quad (3)$$

Here, $\vec{X}_{i,G}$ is the current vector, $\vec{X}_{best,G}$ is the best vector in the neighborhood with respect to the current vector. It's the vector within the corresponding subpopulation for which $\frac{1}{r_{ik}}\left(\frac{f(\vec{X}_{k,G})}{f(\vec{X}_{i,G})}-1\right)$ (k = 1, 2,..., M where M = Number of individuals in the subpopulation and $k \neq i$) is maximum. Here, r_{ik} is the distance between the vectors $\vec{X}_{i,G}$ and $\vec{X}_{k,G}$. It is calculated as $\left\|\vec{X}_{k,G} - \vec{X}_{i,G}\right\|$.

Similarly $\vec{X}_{worst,G}$ is the worst vector in the neighborhood with respect to the current vector. For this vector, $\frac{1}{r_{ik}}\left(1-\frac{f(\vec{X}_{k,G})}{f(\vec{X}_{i,G})}\right)$ (k = 1, 2,..., M where M = Number of individuals in the subpopulation and $k \neq i$) is maximum among all individuals within the subpopulation. Here also r_{ik} has the same meaning as previously.

$\vec{X}_{m,G}$ is the nearest memory individual to $\vec{X}_{i,G}$ in terms of Euclidean distance.

The scaling factors are generated as follows:

$$F^{j}_{m} = 0.3 + 0.7 * rand(0,1) * \left(1 - \frac{|\, x^{j}_{m,G} - x^{j}_{i,G} \,|}{|\, x^{j}_{R} \,|}\right) \quad (4)$$

$$F^{j}_{bw} = 0.3 + 0.7 * rand(0,1) * \left(1 - \frac{|\, x^{j}_{best,G} - x^{j}_{worst,G} \,|}{|\, x^{j}_{R} \,|}\right) \quad (5)$$

where $|\, x^{j}_{R} \,|$ is the search range.

2) Second Stage Mutation: As mentioned earlier, we add the mutant vector from the 1st stage of mutation to the local best vector with a weight factor. In this way we perturb the local best vector in a controlled manner.

The second mutation can be expressed as:

$$\vec{V}_{final,G} = (1-\omega)\cdot\vec{L}_{b,G} + \omega\cdot\vec{V}_{mut,G} \tag{6}$$

Where $\vec{L}_{b,G}$ is the local best vector, i.e. the best vector of the corresponding subpopulation, $\vec{V}_{mut,G}$ is the mutant vector generated from 1st stage mutation and ω is the weight factor.

3.3 Exclusion Rule

For optimizing DOPs, it is important to ensure that the population is almost evenly distributed over the entire search space. If it is not so then the population will lose its exploration capability. In order to achieve the uniform distribution in this multi-population algorithm, we developed an exclusion rule to ensure that different subpopulations are located around different basins of attraction.

Here, the strategy is to calculate the Euclidean distance between each pair of best individuals of two different subpopulations at each generation. If the distance between the best individuals of any two subpopulations falls below the marginal value then the subpopulation having the best individual of lower fitness value among the two is marked for re-initialization. The marginal value of the distance is calculated according to the following rule:

If the search range is R, the search space is D dimensional and there are SP number of subpopulations then the marginal value for the distance is $Dis\_marginal \leq R/(SP*D)$.

3.4 Ageing Mechanism

It may happen that a subpopulation gets trapped in some local optima. This stagnation phenomenon hinders the search process. So, in order to get rid of stagnation at any local optima, we employed a simple aging strategy. Consistent bad performance of any individual is also taken into account in this strategy.

```
Algorithm for Aging Mechanism: Considering i-th individual of j-th
subpopulation. 1. if j-th subpopulation have the global best individual then do not
re-initialize the subpopulation.
2. else if i-th individual is the best individual in j-th subpopulation then
Age_best(j,i)=Age_best(j,i)+1.
if Age_best(j,i)≥30 then re-initialize j-th subpopulation and reset Age_best(j,i) to
0.
3. else if i-th individual is the worst individual in j-th subpopulation then
Age_worst(j,i)=Age_worst(j,i)+1.
if Age_worst(j,i)≥20 then re-initialize the individual and reset Age_worst(j,i) to 0.
    (Initially all the entries of Age_best & Age_worst were 0)
```

3.5 Control Parameter

In order to locate the global optima within the current function landscape, we introduced a control parameter. Depending on the condition this parameter can have a value 0, 1 or 2. We measured the progress of the algorithm every 20 iterations. The control parameter (*C*) is set to the value corresponding to the predefined range in which the progress falls. If *C* becomes 1 then the Quantum individuals are not generated; if *C* becomes 2 then the Brownian individuals are not generated and if *C* becomes 0 then both Quantum and Brownian individuals are generated. Actually, this parameter controls the diversity of the population in order to give high accuracy in locating global optima.

4 Experimental Settings

CEC 2009 benchmark problems for dynamic optimization use the generalized dynamic benchmark generator (GDBG) proposed in [5], which constructs dynamic environments for the location, height and width of peaks. In these benchmark problems, the basic test functions are as follows:

1) F1: Rotation peak function, 2) F2: Composition of Sphere's function, 3) F3: Composition of Rastrigin's function, 4) F4: Composition of Griewank's function, 5) F5: Composition of Ackley's function, 6) F6: Hybrid Composition function.

Only F1 is maximization problem and others are minimization problems. In F1, there are two tests, one using 10 peaks and another using 50 peaks.

There are seven change types for each test functions. However, we tested our algorithm over two most difficult change types for each test function which are large step change and random change. In these change types other algorithms gave the worst results. A detailed description about the experimental settings is given in [4].

5 Parameter Settings

1. Population Size: The Population size (*NP*) was 60. We divided the whole population into 10 subpopulations. Dimensionality was 10 for all the test functions.

2. Control Parameter Adaptation: First we define the difference of the global best fitness values between first 20 iterations as *PR*. From this point onwards, if the global best fitness values between 20 iterations have a difference greater than *PR* then the current value of *PR* is replaced by this new value. If the difference becomes less than (*PR* /10) but greater than (*PR* /50) then *C* is set to 1. If the difference becomes less than (*PR* /50) then *C* is set to 2.

3. Quantum Individual Adaptation: The parameter *R* which is the radius within which the quantum individuals are generated is adaptive in nature. If *C* is 0 then *R* is set to 1. If *C* is 2 then *R* changes according to the following rule depending upon the difference (*Diff*) of global best fitness values between 20 iterations.

$$R = Diff * \log_{10}\left(10 + \frac{PR}{50 * Diff}\right) \tag{7}$$

4. Marginal Value for Distance between Best Individuals: If C is 0 then *Dis_marginal* is set to 0.08 but if C becomes 1 or 2 then *Dis_marginal* becomes 0.03.

5. Crossover Probability: The crossover probability (CR) was kept fixed at 0.9 throughout the search process.

6. Weight Factor for Double Mutation Strategy: The weight factor ω was set to 0.035 for maximization problem (i.e. F1) and 0.1 for minimization problems.

6 Experimental Results

In this section, we represent the performance of our algorithm in terms of mean and standard deviation (STD) of error values obtained for 14 significantly difficult test

Table 1. Comparison of Error Values for Change Type T2

Function	Error Value	Algorithm			
		pDEBQ	DASA	jDE	CESO
F1 (10 Peaks)	Mean	**1.6451e-007**	4.18	3.5874	2.36549
	STD	**2.7157e-007**	9.07	7.83849	7.93215
F1 (50 Peaks)	Mean	**5.0230e-008**	4.86	4.08618	4.35869
	STD	**7.3943e-008**	7.0	6.4546	6.21243
F2	Mean	**4.4666**	25.6	43.004	12.1431
	STD	**7.5912**	83.2	114.944	47.2519
F3	Mean	**750.1274**	824	558.497	791.1642
	STD	**193.5187**	204	384.621	254.1579
F4	Mean	**18.7268**	65.6	49.5044	28.5925
	STD	**25.2722**	160	135.248	98.6229
F5	Mean	**0.16270**	0.99	0.33392	2.282617
	STD	**1.4162**	4.05	1.64364	4.4479
F6	Mean	**9.5653**	37	10.3083	20.0895
	STD	**27.3228**	122	12.2307	60.9701

Table 2. Comparison of Error Values for Change Type T3

Function	Error Value	Algorithm			
		pDEBQ	DASA	jDE	CESO
F1 (10 Peaks)	Mean	**6.2113e-009**	6.37	2.99962	5.17873
	STD	**9.0601e-009**	10.7	7.12954	8.97652
F1 (50 Peaks)	Mean	**0.7390**	8.42	4.29209	6.26213
	STD	**1.7022**	9.56	6.74538	9.16489
F2	Mean	**3.7863**	18.9	50.1906	11.5219
	STD	**6.2560**	67.8	124.015	43.6323
F3	Mean	**750.0162**	688	572.105	634.5213
	STD	**230.8435**	298	386.09	341.2314
F4	Mean	**13.2794**	53.6	51.9448	23.4561
	STD	**23.7620**	140	141.78	92.5379
F5	Mean	**0.29820**	0.949	0.3579	2.81789
	STD	**1.7355**	3.31	1.83299	5.25534
F6	Mean	**10.25390**	26.7	10.954	18.5418
	STD	**30.1322**	98.4	23.2974	65.6901

cases. We have also compared these results with three significant evolutionary DOP solvers - CESO [6], jDE [2], and DASA [7]. We represented it in Table 1& 2.

7 Conclusion

In this paper, a partial differential evolution algorithm namely pDEBQ has been proposed to address DOPs. Key features of the algorithm are briefly outlined as follows:

- It uses a partial DE scheme which obviously shares the traditional DE framework.
- In addition to DE individuals, it uses adaptive Quantum and Brownian individuals to increase the diversity and exploration ability.
- A control parameter is introduced to control the diversity as necessary.
- The algorithm also uses an aging mechanism to get rid of stagnation.
- The DE individuals produce the donor vectors according to a neighborhood based double mutation strategy to control the perturbation.
- An exclusion scheme is used so that the subpopulations become evenly distributed over the entire search space.

The statistical summary of the simulation results also shows that pDEBQ is far better than other algorithms in terms of performance in dynamic landscape. So, we can easily conclude that pDEBQ is very good optimizer for DOPs.

Our future plan includes more co-operation and information exchange between the subpopulations. It will also be valuable to try to make the crossover probability adaptive to the condition of the fitness landscape.

References

1. Storn, R., Price, K.: Differential evolution: A simple and efficient heuristic for global optimizationover continuous spaces. Journal of Global Optimization 11, 341–359 (1997)
2. Brest, J., Zamuda, A., Boskovic, B., Maucec, M.S., Zumer, V.: Dynamic Optimization using Self-Adaptive Differential Evolution. In: Proc. 2009 Cong. on Evol. Comput., pp. 415–422 (2009)
3. Das, S., Suganthan, P.N.: Differential evolution – a survey of the state-of-the- art. IEEE Transactions on Evolutionary Computation 15, 4–31 (2011)
4. Li, C., Yang, S., Nguyen, T.T., Yu, E.L., Yao, X., Jin, Y., Beyer, H.-G., Suganthan, P.N.: Benchmark Generator for CEC 2009 Competition on Dynamic Optimization, Technical Report, University of Leicester, University of Birmingham, Nanyang Technological University (September 2008)
5. Li, C., Yang, S.: A Generalised Approach to Construct Benchmark Problems for Dynamic Optimization. In: Li, X., Kirley, M., Zhang, M., Green, D., Ciesielski, V., Abbass, H.A., Michalewicz, Z., Hendtlass, T., Deb, K., Tan, K.C., Branke, J., Shi, Y. (eds.) SEAL 2008. LNCS, vol. 5361, pp. 391–400. Springer, Heidelberg (2008)
6. Lung, R.I., Dumitrescu, D.: A collaborative model for tracking optima in dynamic environments. In: Proc. 2007 Congr. Evol. Comput., pp. 564–567 (2007)
7. Korosec, P., Silc, J.: The differential anti-stigmergy algorithm applied to dynamic optimization problems. In: Proc. 2009 Congr. Evol. Comput., pp. 407–414 (2009)

PSO Based Memetic Algorithm for Unimodal and Multimodal Function Optimization

Swapna Devi, Devidas G. Jadhav, and Shyam S. Pattnaik

National Institute of Technical Teachers' Training & Research (NITTTR), Sector-26,
Chandigarh, 160019 India
`shyampattnaik@yahoo.com, swapna_devi_p@yahoo.co.in`

Abstract. Memetic Algorithm is a metaheuristic search method. It is based on both the natural evolution and individual learning by transmitting unit of information among them. In the present paper, Genetic Algorithm due to its good exploration capability is used for exploration and Particle Swarm Optimization (PSO) does local search. The memetic process is realized using the fitness information from the individual having best fitness value and searching around it locally with PSO. The proposed algorithm (PSO based memetic algorithm -pMA) is tested on 13 standard benchmark functions having unimodal and multimodal property. When results are compared, the proposed memetic algorithm shows better performance than GA and PSO. The performance of the discussed memetic algorithm is better in terms of convergence speed and quality of solutions.

1 Introduction

Hybridization of Genetic Algorithm with a local search mechanism is known as Memetic Algorithms (MAs) [1]-[3]. It is inspired by both Darwinian principle of natural evolution and Dawkins' notion of a meme as a unit of information or idea transmission from one to another or cultural evolution capable of individual learning [1][2]. In general, MA is a synergy of evolution and individual learning [1] [3], which is improving the capability of evolutionary algorithms (like GA) in finding optimal solutions accurately in function optimization problems with faster speed of convergence.

Stochastic optimizers that have drawn the attentions of researchers in recent times are Genetic Algorithm, Differential Evolution, Evolutionary Programming etc. [4]-[6] where a population of the solutions is utilized in the search process. Many population based algorithms are capable of exploring and exploiting the promising regions in the search space but they take relatively longer time [1]. Hence, algorithms are combined for better exploration and exploitation making it faster as well as accurate. Memetic Algorithm is an outcome of such combination of GA with Local Search. The Real-Coded Memetic Algorithm based on crossover-hill climbing maintains a pair of parents and performs crossover until offspring are obtained for the best offspring selection [3]. Some of the population based search algorithms like Particle Swarm Optimization (PSO) have a tendency of premature convergence [7]. To overcome this drawback, combinatorial optimization method based on particle swarm optimization and stochastic local search is designed in which stochastic search method takes the solution out from local trapping [7]. GA along with gradient-based information as a

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 127–134, 2011.
© Springer-Verlag Berlin Heidelberg 2011

local search has been used to develop memetic algorithms [8]. But the gradient-based methods failed in the cases of multimodal and non-differentiable functions [8]. So, population based local search algorithms have advantages over the gradient type local searches for not getting trapped in local optima [9].PSO is population based and is good for local searching; therefore, it is used as a local search in the proposed algorithm. Genetic Algorithm is mainly used for exploration and the Particle Swarm Optimization (PSO) does the local search. After certain exploratory search by the genetic algorithm, the current best solution obtained is further refined by the local search. The proposed algorithm converges for multimodal functions also. The paper is organized as follows. In section 2 the proposed Memetic Algorithm (pMA) is presented. Section 3 gives the detail about experimentation and results and section 4 concludes the paper.

2 Memetic Algorithm with GA and PSO as Local Search

The GA is used for an intelligent exploration having a random search confined within a defined search space for solving a problem optimally with the help of population [4]. Standard GA applies genetic operators such as selection, crossover, and mutation on initially generated random population in order to evolve a complete generation of new solutions [4] for exploration of the complete search space. GA is good for exploration but it is slow. So, local search is applied which searches for a local optima. This is called memetic algorithm and this enhanced algorithm shows speed of convergence. Elitism prevents the loss of good solutions once the solutions are found [4]. Elitism is used in algorithm for enhancing the performance.

In proposed algorithm, GA is used mainly for exploration purpose and PSO is used for local search. The population is real coded for both genetic algorithm and local search algorithm [3][9]. The population is uniformly distributed random population in the specified range for respective functions as given in Table 1. Based on fitness, the current best individual is selected for elite preservation. Then the population goes through the GA operations viz., crossover and mutation.

The crossovers used are BLX-α crossover [10] and SBX crossover [11]. The offspring $O^1 = \{ o^1_1, o^1_2, \ldots, o^1_i, \ldots, o^1_d\}$ and $O^2 = \{ o^2_1, o^2_2, \ldots, o^2_i, \ldots, o^2_d\}$ are generated from parents $P^1=\{p^1_1, p^1_2, \ldots, p^1_i, \ldots, p^1_d\}$ and $P^2=\{p^2_1, p^2_2, \ldots, p^2_i, \ldots, p^2_d\}$ having 'd' dimension.

Table 1. Benchmark functions used for experimental study (D is the dimension of the functions) [6]

F	Function	Range	NFFE	F	Function	Range	NFFE
F1	Sphere	$[-100,100]^D$	150000	F8	Schwefel's 2.26	$[-500,500]^D$	300000
F2	Schwefel's 2.22	$[-10,10]^D$	200000	F9	Rastrigin	$[-5.12,5.12]^D$	300000
F3	Schwefel's 1.2	$[-100,100]^D$	500000	F10	Ackley	$[-32,32]^D$	150000
F4	Schwefel's 2.21	$[-100,100]^D$	500000	F11	Griewank	$[-600,600]^D$	300000
F5	Rosenbrock	$[-30,30]^D$	500000	F12	Penalized #1	$[-50,50]^D$	150000
F6	Step	$[-100,100]^D$	150000	F13	Penalized #2	$[-50,50]^D$	150000
F7	Quartic	$[-1.28,1.28]^D$	300000				

In BLX-α crossover O is chosen randomly between the interval $[(P^1 - I \cdot \alpha), (P^2 + I \cdot \alpha)]$ with the condition $P^1 < P^2$ and $I = \max(P^1, P^2) - \min(P^1, P^2)$. In SBX crossover the effect of the one-point crossover of the binary representation is tried to emulate. The crossover generates two offspring $O^1=1/2[(1+B) P^1 + (1-B) P^2]$ and $O^2=1/2[(1-B) P^1 + (1+B) P^2]$ where $B \geq 0$ is a sample from random generator having density function shown in equation (1).

$$\begin{aligned} \mathrm{p(B)} &= \frac{1}{2}(\eta+1)\mathrm{B}^{\eta}, \quad if\ 0 \le B \le 1 \\ &\frac{1}{2}(\eta+1)\frac{1}{B^{\eta+2}}, \quad if\ B > 1 \end{aligned} \tag{1}$$

This distribution is obtained by using uniform random number source u(0,1) as shown in equation (2).

$$\begin{aligned} \mathrm{B(u)} &= (2u)^{\frac{1}{\eta+1}}, \quad if\ u \le \frac{1}{2} \\ &\left(\frac{1}{2(1-u)}\right)^{\frac{1}{\eta+1}}, \quad if\ u > \frac{1}{2} \end{aligned} \tag{2}$$

where η=2 is used for better results. Exploitation capacity of operator increases with higher values of η. Mutation changes the variable randomly under uniform normal distribution in the specified range as per Table 1.

The local search algorithm is evoked after a decided number of iterations or generations of genetic algorithm (known as Glocal) are elapsed. A small population is generated around the current best individual by perturbations and given to PSO local search. Velocity and position are updated by equation (3) and (4) respectively.

$$\begin{aligned} v_i^d &= w * v_i^d + c1 * rand1_i^d \left(pbest_i^d - X_i^d\right) \\ &+ c2_i^d * rand2_i^d * \left(gbest^d - x_i^d\right) \end{aligned} \tag{3}$$

$$X_i^d = X_i^d + v_i^d \tag{4}$$

Initially when error is above 0.01 the local search intensity is less *i.e.,* NFFE=200 and when it is below 0.01, intensity is increased by 50% of the previous one. The PSO ensures the particles to be within the range.

The performance of the pMA-BLX-α, pMA-SBX, PSO and GA is checked with 13 standard benchmark functions shown in Table 1. The proposed algorithm is applied on unimodal and multimodal standard benchmark functions and the performance is noted by determining accuracy and convergence speed.

Pseudo Code of MA-PSO is as follows:

```
Initialization: Generate a random initial population
while Stopping conditions are not satisfied do
Evaluate all individuals in the population
Find current best individual (1 elite individual)
If Glocal=2 for best individual do (refinement by PSO)
Generate local population by elite perturbation
While stopping conditions are not satisfied do
Evaluate the population
```

```
Update current_best_individuals, current_Global_best
Calculate velocity and Update the positions
Make the particles feasible as per range, if any
endwhile
Return best individual to main algorithm
end if
Select individuals for crossover with probability $P_c$
Crossover parents by BLX-• or SBX crossover operators
Make the offspring feasible
Mutate individual with mutation probability $P_m$
Replace the parents by offspring preserving the elite
end while
```

3 Experimentation

In the genetic as well as in memetic algorithms, a population of 100 individuals of real-valued representation is used. The crossover operator used is the BLX-α crossover or SBX crossover with crossover probability set to 0.8. For BLX-α crossover, α = 0.3 and for SBX crossover, η =2 are used. The mutation probability is set to 0.05. The algorithm is using generational replacement of individuals preserving one elite. Two stopping criterions are used – the number of fitness function evaluations (NFFE as in Table 1) or an error value of 10^{-8}. PSO is also executed for the same dimensions, population, stopping criterion etc. In PSO, inertia weight is 0.4 and constants considered are c1=c2=2. All functions are having 10 dimensional inputs. All algorithms are executed for 25 independent trials and the best errors in trails are used for averaging and calculating standard deviation (Std Dev) presented in Table 2.

Table 2. Comparison of the experimental results GA, PSO and PSO based Memetic Algorithm (pMA) using PSO as Local Search

F	GA-BLX-α	GA-SBX	PSO	pMA-BLX-α	pMA-SBX
	Mean (Std Dev)	Mean (Std Dev)	Mean (Std Dev)	Mean (Std Dev)	Mean (Std Dev)
F1	5.44E+2 (3.34E+2)	1.57E+3 (4.25E+2)	7.53E-9(1.53E-9)	6.10E-9 (2.27E-9)	6.77E-9 (2.36E-9)
F2	5.86E+0 (4.94E+0)	3.76E+1 (2.16E+1)	6.46E-8 (1.99E-7)	8.80E-09 (1.02E-9)	8.62E-09 (8.77E-10)
F3	9.40E+2 (6.04E+2)	7.10E+3 (2.72E+3)	9.30E-9(7.45E-10)	8.78E-09 (1.13E-9)	9.17E-9 (8.31E-10)
F4	6.86E+0 (1.57E+0)	6.59E+0 (1.88E+0)	9.52E-9 (5.46E-10)	9.30E-09(5.75E-10)	9.28E-09 (5.59E-10)
F5	1.26E+4(1.27E+4)	6.88E+5 (5.03E+5)	3.60E+3 (1.80E+4)	1.95E-1 (8.00E-1)	9.61E-1 (1.74E+0)
F6	5.09E+2(3.36E+2)	1.54E+3 (4.74E+2)	0.00E+0(0.00E+0)	0.00E+0 (0.00E+0)	0.00E+0 (0.00E+0)
F7	1.013E+0 (4.12E-1)	9.84E-1 (4.23E-1)	4.48E-4 (2.32E-4)	3.76E-1 (2.22E-1)	3.30E-1 (2.18E-1)
F8	2.33E+2 (1.49E+2)	5.68E+2 (1.83E+2)	5.67E+2 (1.96E+2)	4.60E+2 (3.49E+2)	6.65E+2 (2.53E+2)
F9	2.39E+1 (5.09E+0)	2.59E+1 (5.69E+0)	2.75E+0(1.36E+0)	1.59E-1 (3.72E-1)	1.59E-1 (3.72E-1)
F10	5.00E+0 (7.83E-1)	4.65E+0 (8.18E-1)	4.62E-2 (2.31E-1)	5.70E-08 (1.98E-8)	6.39E-08 (1.83E-8)
F11	4.03E+0 (1.89E+0)	9.40E+0 (3.11E+0)	6.37E-2(2.98E-2)	1.24E-01 (7.57E-2)	1.47E-1 (7.71E-2)
F12	3.92E+2 (1.45E+3)	1.18E+6 (1.40E+6)	2.07E-2 (4.23E-2)	5.21E-9 (2.10E-9)	5.06E-9 (2.72E-9)
F13	1.14E+4 (3.66E+4)	6.58E+6 (3.99E+6)	8.79E-4 (3.04E-3)	6.22E-9 (2.58E-9)	5.96E-9 (2.50E-9)

Table 3 and Table 4 show the error values achieved by algorithms, pMA-BLX-α and pMA-SBX, respectively in 25 runs for functions F1-F13.

3.1 Discussion

These algorithms are tested on standard benchmark functions having unimodal, multimodal properties with noise and discontinuity. Functions F1-F5 are with unimodal property. Function F6 is a step function, which is having one minimum and is a discontinuous function. Function F7 is a noisy quartic function comprising of random value in range [0, 1). Functions F8 – F13 are multimodal functions in which the number of local minima is increasing exponentially with the problem dimension [6]. For unimodal functions, the convergence rates are more interesting than the final results of optimization. In case of multimodal functions, the final results are important since they reflect algorithm's ability of escaping from poor local optima and locating a good near-global optimum. The pMA is converging quickly in most of the functions like F1-F4, F6, F9-F10, and F12-F13 (both unimodal and multimodal) as shown in Table 3 and Table 4. In Fig. 1 the graphs show the number of fitness function evaluations on x-axis and error in the optimum value averaged over 25 runs on the y-axis. For comparison the convergence

Table 3. Error values achieved by pMA-BLX-α in 25 runs for functions F1-F13

F	1st (Best)	7th	13th (Median)	19th	25th(Worst)	Mean	Std Dev
F1	2.22E-09	4.73E-09	6.23E-09	8.21E-09	9.77E-09	6.10E-09	2.27E-09
F2	6.26E-09	8.08E-09	9.22E-09	9.51E-09	1.00E-08	8.80E-09	1.02E-09
F3	5.69E-09	8.49E-09	9.04E-09	9.47E-09	9.93E-09	8.78E-09	1.13E-09
F4	7.95E-09	8.95E-09	9.48E-09	9.73E-09	9.94E-09	9.30E-09	5.75E-10
F5	3.75E-04	4.98E-04	6.22E-04	8.63E-04	3.99E+0	1.95E-01	8.00E-01
F6	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0
F7	1.21E-01	1.88E-01	3.44E-01	5.22E-01	9.26E-01	3.76E-01	2.22E-01
F8	1.27E-04	2.17E+2	4.34E+2	6.51E+2	1.05E+3	4.60E+2	3.49E+2
F9	3.71E-09	7.55E-09	8.62E-09	9.82E-09	9.95E-01	1.59E-01	3.72E-01
F10	2.79E-08	4.19E-08	5.15E-08	6.90E-08	9.59E-08	5.70E-08	1.98E-08
F11	2.96E-02	7.39E-02	1.01E-01	1.60E-01	3.74E-01	1.24E-01	7.57E-02
F12	1.38E-09	3.60E-09	4.76E-09	6.45E-09	9.49E-09	5.21E-09	2.10E-09
F13	7.78E-10	5.06E-09	6.87E-09	8.28E-09	9.84E-09	6.22E-09	2.58E-09

Table 4. Error values achieved by pMA-SBX in 25 runs for functions F1-F13

F	1st (Best)	7th	13th (Median)	19th	25th(Worst)	Mean	Std Dev
F1	2.51E-09	5.18E-09	7.04E-09	9.07E-09	9.95E-09	6.77E-09	2.36E-09
F2	6.93E-09	8.03E-09	8.65E-09	9.25E-09	9.84E-09	8.62E-09	8.77E-10
F3	7.18E-09	8.86E-09	9.60E-09	9.76E-09	9.98E-09	9.17E-09	8.31E-10
F4	8.15E-09	8.93E-09	9.39E-09	9.80E-09	1.00E-08	9.28E-09	5.59E-10
F5	3.95E-04	6.28E-04	7.66E-04	3.91E-02	4.02E+0	9.61E-01	1.74E+0
F6	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0	0.00E+0
F7	9.03E-02	1.58E-01	3.23E-01	3.66E-01	1.00E+0	3.30E-01	2.18E-01
F8	1.27E-04	5.35E+2	6.91E+2	8.49E+2	1.16E+3	6.65E+2	2.53E+2
F9	4.90E-09	7.88E-09	8.98E-09	9.79E-09	9.95E-01	1.59E-01	3.72E-01
F10	3.65E-08	5.03E-08	6.40E-08	7.52E-08	9.80E-08	6.39E-08	1.83E-08
F11	3.74E-02	9.59E-02	1.21E-01	1.99E-01	3.17E-01	1.47E-01	7.71E-02
F12	1.62E-09	2.71E-09	4.70E-09	6.96E-09	9.50E-09	5.06E-09	2.72E-09
F13	1.47E-09	3.54E-09	6.13E-09	7.92E-09	9.37E-09	5.96E-09	2.50E-09

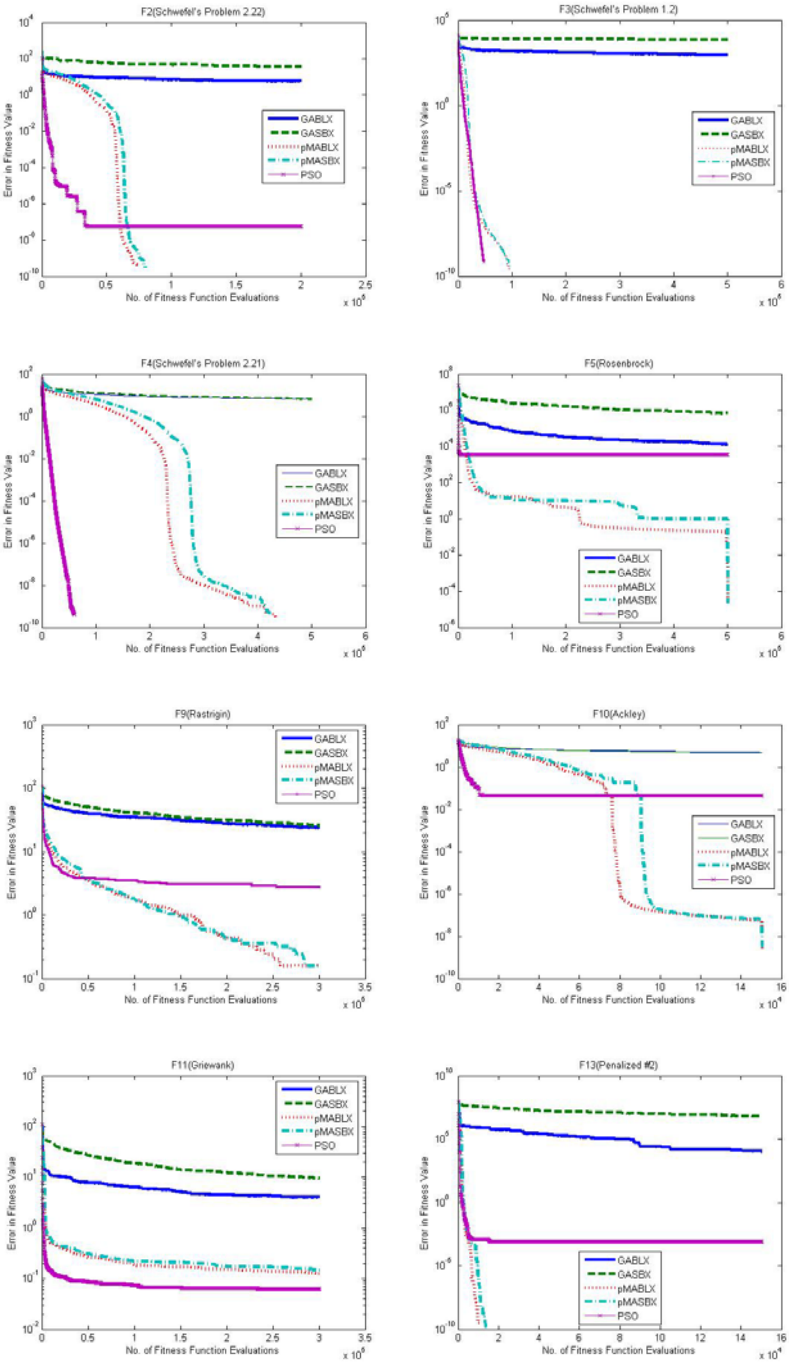


Fig. 1. Average fitness error versus NFFE for functions F2, F3, F4, F5, F9, F10, F11& F13

characteristics of GA-BLX-α, GA-SBX, pMA-BLX-α, pMA-SBX and PSO are plotted. As seen from the figure, the GA-BLX-α and GA-SBX are performing in a similar way and pMA-BLX-α and pMA-SBX are also having similar nature of convergence. PSO is converging better than GA but showing lower performance than pMA. The future work could be to integrate local search with improved PSO variants [12] and incorporating CEC 2005 test problems [13] for validation.

4 Conclusion

In this paper a new variant of memetic algorithm named pMA is developed. Particle Swarm Optimization (PSO) is used as the local search in this Memetic Algorithm. The mechanism of PSO for dragging all the particles towards the best one shows the instating property of 'meme'. PSO is good at local search and showing its usefulness in the pMA. The proposed PSO based Memetic Algorithm (pMA) is performing robustly as compared to Genetic Algorithms in terms of higher accuracy and convergence speed. PSO is also showing promising performance individually and supporting GA very well to make the MA. Both the algorithms are having convergence accuracy upto the error value 10^{-8} for most of the functions and showing better speed, success and consistency of the convergence. The variants of PSO can be integrated as a local search for making MA and can be tested on CEC2005 test problems.

References

1. Nguyen, Q.H., Ong, Y.S., Krasnogor, N.: A Study on the Design Issues of Me-metic Algorithm. In: Proc. of the IEEE Congr. Evol. Comput. (CEC 2007), pp. 2390–2397 (September 2007)
2. Moscato, P.A.: On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms, Tech. Rep. Caltech Concurrent Computation Program, California Institute of Technology, Pasadena, CA, Report 826 (1989)
3. Lozano, M., Herrera, F., Krasnogor, N., Molina, D.: Real-Coded Memetic Algo-rithms with Crossover Hill-Climbing. Evolutionary Computation 12(3), 273–302 (2004)
4. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Springer, Heidelberg (1996)
5. Das, S., Suganthan, P.N.: Differential evolution - a survey of the state-of-the-art. IEEE Trans. on Evolutionary Computation 15(1), 4–31 (2011)
6. Yao, X., Liu, Y., Lin, G.: Evolutionary Programming Made Faster. IEEE Trans. on Evolutionary Computation 3(2), 82–102 (1999)
7. Akbari, R., Ziarati, K.: Combination of Particle Swarm Optimization and Stochastic Local Search for Multimodal Function Optimization. In: Proc. of the IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application (PACIIA 2008), pp. 388–392 (2008)
8. Li, B., Ong, Y.S., Le M.N., Goh, C.K.: Memetic Gradient Search. In: Proc. of the IEEE Congress on Evol. Comput. (CEC 2008), pp. 2894–2901 (2008)
9. Jadhav, D.G., Pattnaik, S.S., Devi, S., Lohokare, M.R., Bakwad, K.M.: Approximate Memetic Algorithm for Consistent Convergence. In: Proc. National Conf. on Computational Instrumentation (NCCI 2010), pp. 118–122 (March 2010)

10. Eshelman, L.J., Schaffer, J.D.: Real-coded genetic algorithms and interval-shemata. In: Darrell Whitley, L. (ed.) Foundation of Genetic Algorithms, vol. 2, pp. 187–202. Morgan Kaufmann, San Mateo (1993)
11. Deb, K., Agrawal, R.B.: Simulated binary crossover for continuous search space. Complex Syst. 9(2), 115–148 (1995)
12. Liang, J.J., Qin, A.K., Suganthan, P.N., Baskar, S.: Comprehensive Learning Particle Swarm Optimizer for Global Optimization of Multimodal Functions. IEEE Trans. on Evol. Comput. 10(3), 281–295 (2006)
13. Suganthan, P.N., Hansen, N., Liang, J.J., Deb, K., Chen, Y.P., Auger A., Tiwari, S.: Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization. Technical Report, Nanyang Technological University, Singapore, & KanGAL Report #2005005, IIT Kanpur, India (May 2005)

Comparison of PSO Tuned Feedback Linearisation Controller (FBLC) and PI Controller for UPFC to Enhance Transient Stability

M. Jagadeesh Kumar[1], Subranshu Sekhar Dash[2], M. Arun Bhaskar[3],
C. Subramani[4], and S.Vivek[5]

[1,3,5] Dept. of EEE, Velammal Engineering College, Chennai-66
[2,4] Dept. of EEE, SRM University, Kattankulathur, Chennai
mp.jagadeesh@gmail.com

Abstract. An Unified Power Flow Controller (UPFC) is a typical Flexible AC Transmission System (FACTS) device playing a vital role as a stability aid for large transient disturbances in an interconnected power system. This paper deals with the design of Feedback Linearising Controller (FBLC) tuned by PSO for UPFC.The disturbances are created in the SMIB system and the results are simulated. The results proved the supremacy of the Power System Stabilizer (PSS) equipped with FBLC over the PSS equipped with Proportional Integral (PI) controller.

Keywords: UPFC, PSO, FACTS, FBLC, PSS, PI.

1 Introduction

1.1 Transient Stability

Transient stability refers to ‘‘the ability of a power system to maintain synchronism when subjected to a severe transient disturbance’’ [1]. Stabilization of a synchronous generator is undoubtedly one of the most important problems in power system control. Power system stabilizers (PSS) and Automatic voltage regulators (AVR) with exciter are normally employed to damp out the electromechanical oscillations as well as for the post fault bus-voltage recovery. However, it is well known that the performances of PSS and AVR are limited since their designs are primarily based on linear control algorithms. In the event of large faults, the nonlinearities of the system become very severe, thereby putting limitations on the performances of classical control designs. Thus, the most appropriate approach for controller design for a power system is the use of nonlinear control theory, i.e., multivariable feedback linearization scheme. The application of feedback linearization approaches for power system control was first investigated by Marino [2] and subsequently by several researchers [3-5]. This control technique has also been successfully applied to control of drives and power electronics based systems [6-8]. Successful applications of FACTS equipment for power flow control, voltage control and transient stability improvement have been reported in the literature [9-13].The rapid development of power electronics technology provides opportunities to develop new power equipment to improve the performance of the actual power systems. During the last decade, a technology called “Flexible AC Transmission Systems” (FACTS) have been proposed and implemented.

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 135–142, 2011.
© Springer-Verlag Berlin Heidelberg 2011

2 Modeling of UPFC

2.1 UPFC Installed in SMIB System

The mathematical model of the UPFC is derived here in the d-q (synchronously rotating at the system angular frequency ω) frame of reference. This is followed by a detailed description of the conventional PI control strategy employed for active and reactive power control using UPFC. The equivalent circuit model of a power system equipped with a UPFC is shown in Fig 2. The series and shunt VSIs are represented by controllable voltage sources V_c and V_p, respectively. R_p and L_p represent the resistance and leakage reactance of the shunt transformer respectively. Leakage reactance and resistance of series transformer have been neglected.

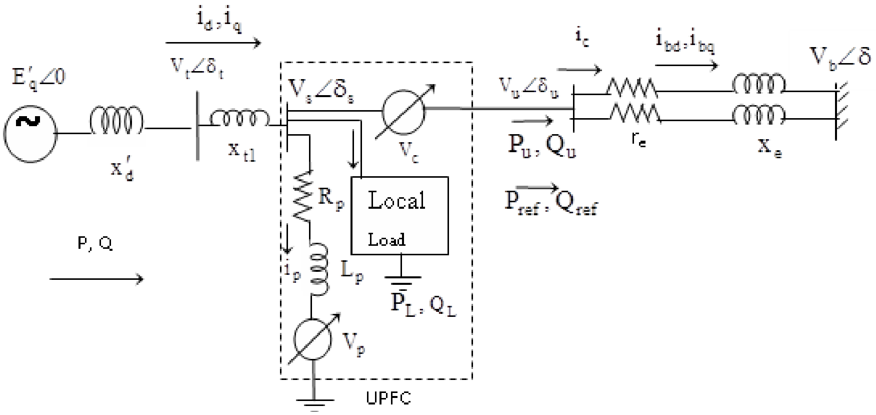


Fig. 1. Single line diagram of UPFC installed in power system

2.2 Modeling of Shunt Converter

$$\left[R_p + L_p\frac{d}{dt}\right] i_p = V_s - V_p \tag{1}$$

$$\frac{di_p}{dt} = -R_p\frac{i_p}{L_p} + \frac{1}{L_p}\left(V_s - V_p\right) \tag{2}$$

Where:

$$i_p = \left[i_{pa} i_{pb} i_{pc}\right]^T$$
$$V_s = \left[V_{sa} V_{sb} V_{sc}\right]^T$$
$$V_p = \left[V_{pa} V_{pb} V_{pc}\right]^T$$
$$R_p = \begin{bmatrix} R_p & 0 & 0 \\ 0 & R_p & 0 \\ 0 & 0 & R_p \end{bmatrix} \text{and} L_p = \begin{bmatrix} L_p & 0 & 0 \\ 0 & L_p & 0 \\ 0 & 0 & L_p \end{bmatrix}$$

Under the assumption that the system has no zero sequence components, all currents and voltages can be uniquely represented by equivalent space phasors and then transformed into the synchronous *d-q-o* frame by applying the following transformation (q is the angle between the *d*-axis and reference phase axis):

$$T = \sqrt{\frac{2}{3}} \begin{bmatrix} cos\theta & \cos\left[\theta - \frac{2\pi}{3}\right] & \cos\left[\theta + \frac{2\pi}{3}\right] \\ -\sin\theta & -\sin\left[\theta - \frac{2\pi}{3}\right] & -\sin\left[\theta + \frac{2\pi}{3}\right] \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$$

Thus, the transformed dynamic equations are given by,

$$\frac{\mathrm{di}_{\mathrm{pd}}}{\mathrm{dt}} = -\mathrm{R}_{\mathrm{p}} \frac{\mathrm{i}_{\mathrm{pd}}}{\mathrm{L}_{\mathrm{p}}} + \frac{1}{\mathrm{L}_{\mathrm{p}}} \left(\mathrm{V}_{\mathrm{sd}} - \mathrm{V}_{\mathrm{pd}}\right) + \omega \mathrm{i}_{\mathrm{pq}} \tag{3}$$

$$\frac{\mathrm{di}_{\mathrm{pq}}}{\mathrm{dt}} = -\mathrm{R}_{\mathrm{p}} \frac{\mathrm{i}_{\mathrm{pq}}}{\mathrm{L}_{\mathrm{p}}} + \frac{1}{\mathrm{L}_{\mathrm{p}}} \left(\mathrm{V}_{\mathrm{sq}} - \mathrm{V}_{\mathrm{pq}}\right) - \omega \mathrm{i}_{\mathrm{pd}} \tag{4}$$

where ω is the angular frequency of the AC bus voltage.

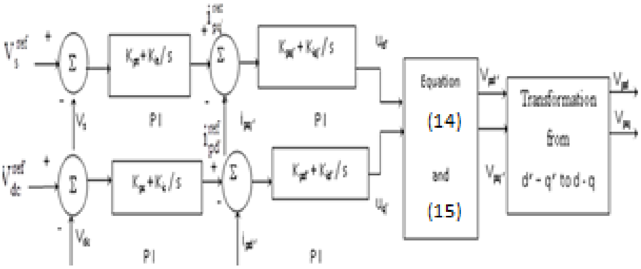


Fig. 2. PI-Control Structure of Shunt Converter

2.3 Modeling of Series Converter

$$\frac{di_{bd}}{dt} = -\frac{\omega_b r_e}{x_e} i_{bd} + \omega i_{bq} + \frac{\omega_b}{x_e} (V_{ud} - V_b \sin\delta) \tag{5}$$

$$\frac{di_{bq}}{dt} = -\frac{\omega_b r_e}{x_e} i_{bq} - \omega i_{bd} + \frac{\omega_b}{x_e} \left(V_{uq} - V_b \cos\delta\right) \tag{6}$$

For fast voltage control, the net input power should instantaneously meet the charging rate of the capacitor energy. Thus, by power balance,

$$\begin{aligned} P_s - P_u &= V_{sd}\left(i_{pd} + i_{bd}\right) + V_{sq}\left(i_{pq} + i_{bq}\right) - (V_{ud} i_{bd} + V_{uq} i_{bq}) \\ &= \mathrm{V}_{\mathrm{dc}} \mathrm{i}_{\mathrm{dc}} \\ &= V_{dc}\left(C \frac{dV_{dc}}{dt} + gV_{dc}\right) \end{aligned}$$

$$\frac{dV_{dc}}{dt} = -\frac{g\omega V_{dc}}{b} + \frac{1}{CV_{dc}} \left[V_{sd} i_{pd} + V_{sq} i_{pq} + (V_{sd} - V_{ud}) i_{bd} + \left(V_{sq} - V_{uq}\right) i_{bq}\right] \tag{7}$$

An appropriate series voltage (both magnitude and phase) should be injected for obtaining the commanded active and reactive power flow in the transmission line, i.e.,

(P_u,Q_u) in this control. The current references are computed from the desired power references and are given by,

$$i_{cd}^{ref} = \frac{P_{ref}V_{ud} - Q_{ref}V_{uq}}{V_u^2} \tag{8}$$

$$i_{cq}^{ref} = \frac{P_{ref}V_{uq} - Q_{ref}V_{ud}}{V_u^2} \tag{9}$$

The corresponding control system diagram is shown in Fig 6 .

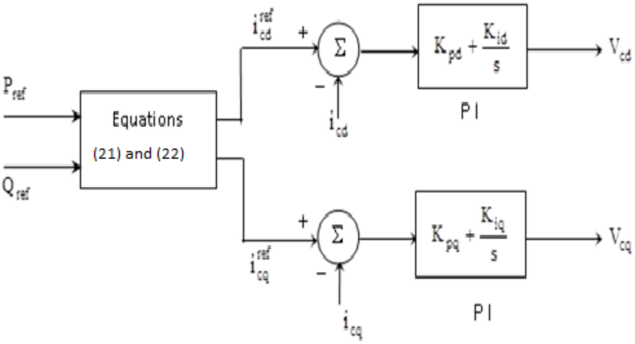


Fig. 3. PI-Control of Series Converter

3 Particle Swarm Optimisation

PSO is basically developed through simulation of bird flocking in a two-dimension space. The position of each agent is represented by its XY-axis position and the velocity is expressed by V_X (the velocity of x-axis) and V_Y (the velocity of y-axis). Modification of the agent position is realized by position and velocity information. Bird flocking optimizes a certain object function. Each agent knows its best value (pbest) and its XY position. This information is an analogy of personal experiences of each agent. Moreover, each agent knows the best value in the group (gbest) among bests.

This information is an analogy of knowledge of how the other agents around them have performed. Namely, each agent tries to modify its position.

- The current position (X,Y)
- The current velocities(V_X, V_Y)
- The distance between the current position and pbest
- The distance between the current position and gbest

$$v(k+1)_{i,j} = w.v(k)_{i,j} + c_1r_1(gbest - x(k)_{i,j}) + c_2r_2(pbest_j - x(k)_{i,j})$$

$$x(k+1)_{i,j} = x(k)_{i,j} + v(k)_{i,j}$$

where

$v_{i,j}$ velocity of particle i and dimension j, $x_{i,j}$ position of particle i and dimension j c_1,c_2 Acceleration constants, w inertia weight factor, r_1,r_2 Random numbers between 0 and 1, pbest Best position of a specific particle, gbest Best particle of the group

4 FBLC for UPFC

In this section, the design steps for the feedback linearizing control of UPFC have been presented. In the UPFC control, there are two objectives, i.e. ac bus voltage (Vdc) control. in the following control design, the dynamic equation for Vs is obtained as follows with reference to single line diagram. Now, for the control design, the complete state space model is expressed in the form of equations as follows:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} i'_{pd} \\ i'_{pq} \\ i'_{bd} \\ i'_{bq} \\ v_{dc} \\ v_s \end{bmatrix} \qquad \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} v'_{pd} \\ v'_{pq} \\ v'_{cd} \\ v'_{cq} \end{bmatrix} \qquad \begin{aligned} x_1 &= i'_{pd} \\ x_2 &= i'_{pq} \\ x_3 &= i'_{bd} \\ x_4 &= i'_{bq} \\ x_5 &= v_{dc} \end{aligned}$$

$$\begin{aligned} \dot{x}_1 &= f_1(x) + g_1 u_1 \\ \dot{x}_2 &= f_2(x) + g_2 u_2 \\ \dot{x}_3 &= f_3(x) + g_3 u_3 \\ \dot{x}_4 &= f_4(x) + g_4 u_4 \\ \dot{x}_5 &= f_5(x) \end{aligned}$$

$$\dot{x}_6 = f_6(x) + g_{61}u_{61} + g_{62}u_{62} + g_{63}u_{63} + g_{64}u_{64}(x)$$

The outputs of the system are *Vs* and *Vdc*. thus,

$$\begin{gathered} y_1 = i_{dse} \\ y_2 = i_{qse} \\ y_3 = v_s \\ y_4 = v_{dc} \end{gathered}$$

By control law, the outputs of the system is given by,

$$\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \\ \ddot{y}_3 \\ \dot{y}_4 \end{bmatrix} = \begin{bmatrix} f3(x) \\ f4(x) \\ f6(x) \\ f5(x) \end{bmatrix} + \begin{bmatrix} 0 & 0 & g_3 & 0 \\ 0 & 0 & 0 & g_4 \\ g_{61} & g_{62} & g_{63} & g_{64} \\ g_{51} & g_{52} & g_{53} & g_{54} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix}$$

$$= A(x) + E(x) \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix}$$

The non-singularity of ***E(x)*** can be observed by computing the determinant of ***E(x)***.

E(x) is non-singular in the operating ranges of *Vs* and *Vdc*. For tracking of *Vs* and *Vdc*, the new control inputs *v1, v2, v3* and *v4* are selected as (by both proportional and integral control):

$$\begin{pmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{pmatrix} = \begin{pmatrix} \dot{y}_{1ref} + K_{11}e_1 + K_{12}\int e_1 dt \\ \dot{y}_{2ref} + K_{11}e_2 + K_{12}\int e_2 dt \\ \ddot{y}_{3ref} + K_{11}e_3 + K_{12}\int e_3 dt \\ \dot{y}_{4ref} + K_{21}\dot{e}_4 + K_{22}e_4 + K_{23}\int e_4 dt \end{pmatrix}$$

The gain parameters K_{11}, K_{12}, K_{21}, K_{22},$K_{23}$are determined by assigning desired poles on the left half s-plane and, thus, asymptotic tracking control to the reference can be achieved.

5 Simulation Results

5.1 CASE-I

The synchronous generator is assumed to operate at P=1.2p.u.and Q = 0.85 p.u. A 3-phase fault occurs near the infinite bus for duration of 100 ms The 3-phase fault is created at 500ms and removed at 600ms.

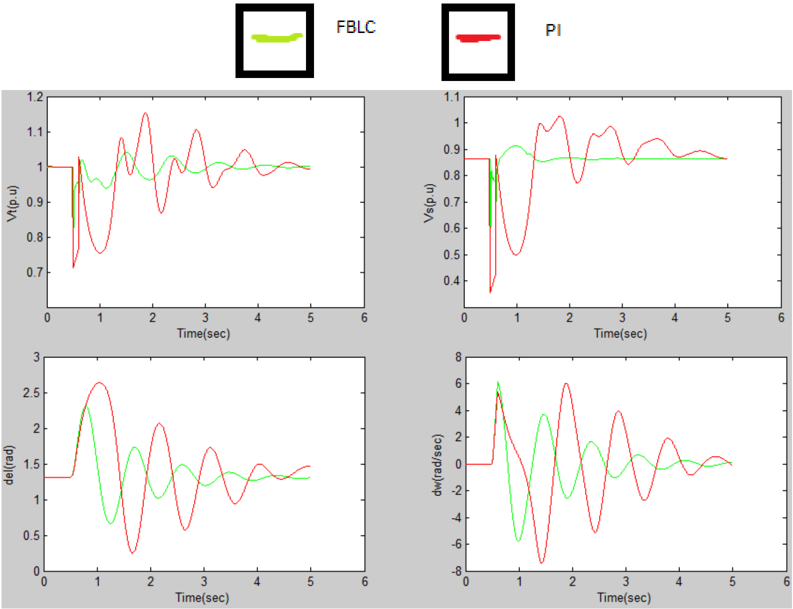


Fig. 4. Comparison of transient responses between FBLC and PI-Controller

5.2 CASE-II

The synchronous generator is assumed to operate at P=1.2p.u.and Q = 0.5 p.u. A 3-phase fault occurs near the infinite bus for duration of 100 ms The 3-phase fault is created at 500ms and removed at 600ms.

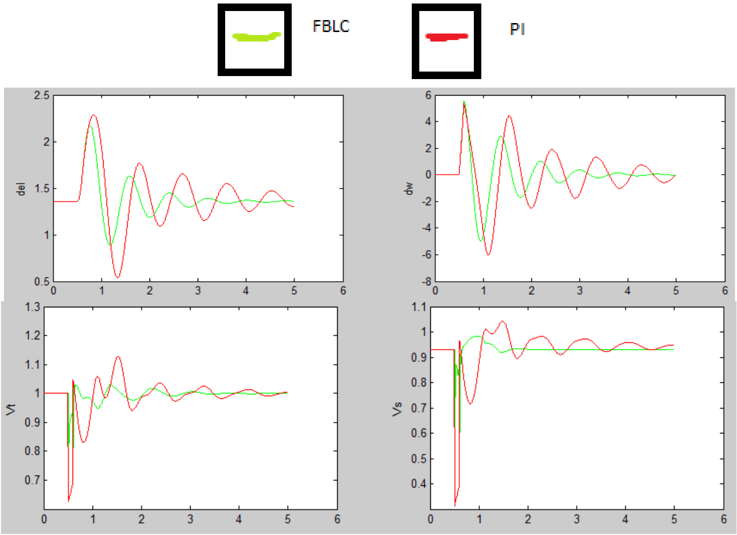


Fig. 5. Comparison of transient responses between FBLC and PI-Controller

6 Conclusion

The mathematical model for Unified Power Flow Controller (UPFC) has been developed and the MATLAB experimental results proved the supremacy of the Feed-Back Linearization Controller (FBLC) over the ordinary Proportional Integral (PI) Controller. For all type of disturbances created in the Single Machine Infinite Bus (SMIB) system, the Feed-Back Linearization Controller (FBLC) equipped with Power System Stabilizers (PSS) damped the electromechanical oscillations faster than the Proportional Integral (PI) controller equipped with Power System Stabilizers (PSS).

References

[1] Anderson, P.M., Fouad, A.A.: Power System Control and Stability. IEEE Press (1994)

[2] Akhrif, O., Okou, F.A., Dessaint, L.A., Champagne, R.: Application of a multivariable feedback linearization scheme for rotor angle stability and voltage regulation of power systems. IEEE Transactions on Power Systems 14(2), 620–628 (1999)

[3] Tan, Y.L., Wang, Y.: Design of series and shunt FACTS controller using adaptive nonlinear coordinated design techniques. IEEE Transactions on Power Systems 12(3), 1374–1379 (1997)

[4] Gyugyi, L.: Dynamic compensation of AC transmission lines by solid-state synchronous voltage sources. IEEE Transactions on Power Delivery 9, 904–911 (1994)

[5] Mihaliæ, R., Zunko, P., Papiæ, I., Povh, D.: Improvement of transient stability by insertion of FACTS devices. In: IEEE/NTUA Athens Power Tech. Conference Proc., pp. 521–525 (1993)

[6] Saravanailango, G., Nagamani, C.: A non-linear control technique for UPFC based on linearisation. International Journal of Electric Power Components and Systems 36, 432–447 (2008)

A Nelder-Mead PSO Based Approach to Optimal Capacitor Placement in Radial Distribution System

Pradeep Kumar and Asheesh K. Singh

Electrical Engineering Department, MNNIT Allahabad, India
pradeepkumar0802@gmail.com, asheesh_k_singh@yahoo.com

Abstract. In distribution system, the size and location of shunt capacitors for peak power loss and energy loss reduction plays a vital role. This paper proposes a new method, Nelder-Mead particle swarm optimization (NM-PSO) for the optimal capacitor placement problem. The NM-PSO method is applied to IEEE 69-bus radial distribution system and the results obtained are compared with that of particle swarm optimization (PSO). Also, the problem has been reformulated to include the maintenance cost and economic factors such as inflation and interest rates. The results obtained clearly indicate the better performance of NM-PSO over PSO, confirming its superiority to PSO in finding the optimal solution and handling more complex, nonlinear objective functions.

1 Introduction

The power system aims to feed the customer loads with quality electrical energy as economical as possible. Distribution system, one of the three components of power system is responsible for energy transfer to these electrical loads. Studies indicate that in a power system major portion of the total system losses, nearly 13% occur in distribution system [1]. With development, the increase in quantity of electrical loads will make the system more inefficient by increasing the distribution system losses. Therefore, the utilities need to increase the efficiency of the distribution system.

To reduce these power losses the utilities provide the reactive power demands locally by shunt capacitors. Due to their economical cost and high cost benefits shunt capacitors have been the preferred choice of the utilities. However, the amount of the profit that can be achieved using these capacitors largely depends upon their optimal location and size. Investment cost is one of the most important factors for optimal capacitor placement (OCP) amongst other factors such as cost of planning, operation, design process investment, power or energy etc. Thus, economic consideration becomes necessary for obtaining the best solution in capacitor placement design. This problem of OCP has attracted many researchers, taking the problem's publication count to over 400 [1]. Various heuristic methods [2, 3] and Numerical programming [4] are also used for solving the problem. Recently, artificial intelligence based methods [5-8] have also been used to optimally placing the capacitors.

Researchers have modeled the OCP problem as different objective functions. In some papers, maximizing dollar saving function [3], [5-9] remains the objective while in some it remains only reducing the power losses [10]. But these objective functions assume the

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 143–150, 2011.
© Springer-Verlag Berlin Heidelberg 2011

costs to remain constant over the planning period, without including the maintenance cost, market inflation and creditor's interest. This paper proposes a new optimization algorithm of evolutionary computation, known as Nelder-Mead particle swarm optimization (NM-PSO) [10], for solving OCP problem. NM-PSO combines the advantages of both Nelder-Mead (NM) and particle swarm optimization (PSO) algorithms.

A new objective function of the problem with the objectives of cost reduction of peak power loss, energy loss and capacitor's cost with effects of maintenance cost, inflation and interest rates on cost. The candidate bus selection is done using the bus sensitivity method [12]. The NM-PSO algorithm is applied to the IEEE 69-bus radial distribution system and the results are compared with PSO algorithm. The results obtained show the applicability of NM-PSO over PSO for OCP. The results show that with NM-PSO mores savings can be achieved with less kVArs placed in the system in comparison to PSO.

2 PSO and NM-PSO

This section describes the PSO and NM-PSO briefly. The PSO algorithm is an evolutionary computation technique, imitating the social behavior amongst entities. In NM-PSO, the efficient local direct search technique is combined with PSO. This helps PSO is searching the space effectively.

2.1 Method 1: Particle Swarm Optimization (PSO)

PSO [13, 14] is an evolutionary computation technique imitating the social behavior of bird flocking and fish schooling. It is a population based stochastic algorithm, where population and individual entities are termed as *swarm* and *particles*, respectively. These particles move through the problem hyperspace with given velocity. In PSO, particles in the swarm update their positions stochastically towards the successful regions. The success of particles is influenced by particle's previous best position and success of their topological neighbors. The detailed analysis of this algorithm is presented in [13, 14].

For an *n*-dimensional search space the position ($\overline{x_i(t)}$) and velocity ($\overline{v_i(t)}$) of particles, at any instant *t*, can be given as:

$$\overline{x}_i(t) = [x_1(t), x_2(t), x_3(t), \ldots, x_n(t)]^T \tag{1}$$

$$\overline{v}_i(t) = [v_1(t), v_2(t), v_3(t), \ldots, v_n(t)]^T \tag{2}$$

Keeping the relative importance of particle's own and neighborhood experiences, the updated velocity of the particle is given as

$$\overline{v_i(t+1)} = \overline{v_i(t)} + \varphi_1.rand_1.(p_{best} - x_i(t) + \varphi_2.rand_2.(g_{best} - x_i(t)) \tag{3}$$

where, φ_1, φ_2 are two positive numbers (1.5, 2), $rand_1$, $rand_2$ are two random numbers [0, 1], p_{best} is local best of particles, g_{best} is global best of particles. $\overline{v_i(t)}$ is the velocity of the particles taken as difference between maximum and minimum capacitor values divided by number of iteration respectively. Accordingly, the updated position of the particle is given as

$$\bar{x}_i(t+1) = \bar{x}_i(t) + \bar{v}_i(t+1) \tag{4}$$

The pseudo code of the above algorithm is shown in Fig.1.

```
1. Initialize each particle
   End
2. Do
     For each particle
        Calculate fitness value
         If the fitness value is better than the best fitness value ($p_l$)
            set current value as the new $p_l$
   End
     Choose the particle with the best fitness value of all the particles as the $p_g$
       For each particle
          Calculate particle velocity according equation (3)
          Update particle position according equation (4)
   End
```

Fig. 1. Pseudo code of particle swarm optimization (PSO)

2.2 Method 2: Nelder-Mead Particle Swarm Optimization (NM-PSO)

The hybrid NM-PSO algorithm, as proposed by Fan and Zahara [10], is based on combination of NM method and PSO for constrained optimization. Nelder and Mead proposed a simple local direct search technique, which does not require any derivative for finding solution of any function [15]. The PSO is used as global search technique but it is limited by high computational cost of the slow convergence rate.

The slow convergence rate of PSO than NM is due to improper utilization of local information to determine a most promising search direction. To overcome this slow convergence PSO is combined with NM, in a way that both the algorithms enjoy merits of each other. In NM-PSO, PSO prevents the hybrid approach from getting trapped in local optima, whereas NM increases the convergence rate. Summarily, PSO focuses on "exploration" and NM method focuses on "exploitation" [11].

For an *N*-dimensional problem, the pseudo code of the algorithm is given in Fig. 2.

```
1. Initialize. Generate a population of size 3N+1
   Repeat
2. Evaluate the fitness of each particle
   Rank them based on the fitness results.
3. Simplex Method. Apply NM operator to top N+1 particle.
4. PSO method. Apply PSO operator to remaining 2N particles, and update the
   particles with worst fitness until stopping criterion is reached
```

Fig. 2. Pseudo code for Nelder-Mead particle swarm optimization (NM-PSO)

There are various parameters to be specified for the experiments. The parameters for PSO have already been discussed in previous sub-section. The various parameters of the NM algorithm in NM-PSO viz. reflection coefficient (α), expansion coefficient (β), Contraction coefficient (γ), and Shrink Coefficient (δ) are given as 1, 2, 0.5, and 0.5, respectively. The experiments are performed for 100 iterations.

3 Bus Sensitivity

Sensitivity of any particular bus can be calculated using bus sensitivity method [12], which reveals that reactive power compensation is mainly required on buses having reactive power demand and below nominal bus voltage. The index bus sensitivity for each bus *i* bus sensitivity (BS_i) index is defined as

$$BS_i = \lambda_{Qi} \frac{Ql_i}{V_i} \tag{5}$$

where, λ_{Qi}, Ql_i, and V_i are the Lagrange multiplier, load reactive power and voltage magnitude at bus *i*, in that order. For the capacitor placement problem, λ_{Qi} imitates the sensitivity of objective function towards the changes in the reactive power injection to bus bar i. Bus sensitivity index BS_i gives cost required to increase the voltage at bus *i*. The unit of BS_i is \$/V.

4 Problem Formulation

The objectives of OCP are to minimize power loss, energy loss, and cost of capacitors, along with maintaining the bus voltages in a limit. The costs included in the objective function are affected by the maintenance cost and the economic factors. The reformulated objective function incorporating these costs and factors, can be expressed as

$$\min f(u^0) = k_e \sum_{i=1}^{nl} T_i P_i + k_p P_l + \sum_{j=1}^{nc} C_j \left(u_j^0\right) k_{ef}^Y + \sum_{Y=1}^{L_c} \left(M_{tc} (1+I)^{Y-1}\right) \tag{6}$$

where, T_i is time duration, nl is load level, $C_j(u^0_j)$ is cost function of capacitor $C_j(u_j^0) = k_{inst} + k_{cj} u_j^0$, nc is number of capacitors, P_l is power loss at peak load, k_{ef} is economic factor given as $k_{ef} = \frac{1+m}{1+f}$, and P_i power loss at load level *i*. The values for capacitor cost (k_{cj}), capacitor installation cost (k_{inst}), cost of energy (k_e), cost of peak power loss (k_p), Planning period (Y), maintenance cost (M_{tc}), increment (I) in M_{tc}, inflation rate (m), and interest rate (f) are taken as US\$ 3/kVAr, US\$ 1000, US\$ 0.06/kWh, US\$ 168/kW, 10 years, US\$ 322.42/year, 10% per year, 4% per year, and 5% per year, respectively.

Initially, based on distribution load flow [16], the power losses for the system are calculated without any capacitor placement. BS_i calculation forms the basis of bus selection for capacitor placement. Then at the selected buses the capacitors are placed utilizing the two methods (i.e. PSO and NM-PSO) and the objective function. For the given system the process of determination of the OCP is done till the minimum value of the objective function is obtained following the constraints. Performance of the system is then evaluated based on the solutions obtained.

4.1 Operating Constraints

The problem of capacitor placement is subjected to the following constraints:

(1). **AC power flow constraints.** The power flow constraint as be expressed as:

$$G^i(x^i, u^i) = 0; \quad i = 1,2,...,nl \tag{7}$$

(2). **Bus voltage constraint.** The bus voltages should lie in between the minimum, V_{min} (0.9 pu) and V_{max} (1.1 pu).

$$V_{\min} < V < V_{\max} \tag{8}$$

(3). **Capacitor capacity constraint**. The maximum number of capacitors to be installed in a system is limited, i.e.

$$\frac{u_j^0}{u_s} \in \{1,2,\ldots,nc\} \tag{9}$$

5 Results and Discussions

The proposed objective function and algorithm, has been implemented on the MATLAB platform. It has been implemented on 12.6 kV IEEE 69-bus system [17] radial distribution system. The diagram of the test feeder system is shown in Fig. 3. Load duration data for the IEEE 69-bus system is given in Table 1.

5.1 Candidate Bus Selection

Bus sensitivity method is adopted for candidate bus selection. Buses having highest numeric value of BS_i are selected for capacitor placement. The bus sensitivity calculation for IEEE 69-bus system, on this basis 9 buses are selected for capacitor placement, i.e. 11, 12, 21, 48, 49, 50, 59, 61, and 64. Based on the capacitors selected the problem of OCP becomes a 9 (N=9) dimensional problem.

5.2 Capacitor Placement at Buses

The capacity of capacitors placed on selected buses of test system is calculated using method-1 and method-2. Table 2 shows the capacitors placed at the candidate buses using the two methods. After placement of capacitors the performance of the bus

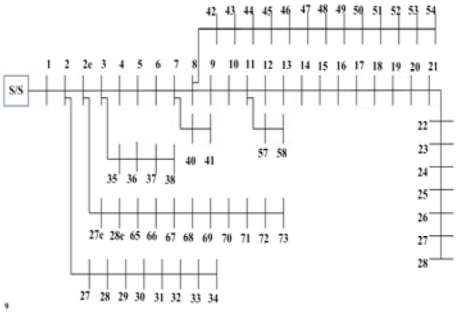


Fig. 3. IEEE 69-bus radial distribution system

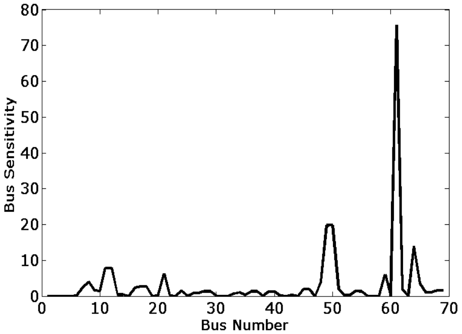


Fig. 4. Bus sensitivity of IEEE 69-bus radial distribution system

Table 1. Load Duration Data for IEEE 69-bus System

	Low	Medium	Peak
Load Level	0.5	1	1.4
Time Interval (Hrs)	1000	6760	1000

systems is evaluated by considering peak power loss, energy loss, total cost and the benefits. The results obtained are tabulated in Table 3.

From Table 2 and 3 it is clearly visible that with method-1 the OCP results in considerable reduction in peak power loss of 29.58%. Also, the cost energy losses are reduces to US$ 220,445.3 from the original US$ 305513.9. This results in yearly benefits of US$ 95, 820.17, which is nearly 19.22%. Method-2 applied for capacitor placement also provides 29.58% peak power loss reduction, but the cost of energy losses reduces from US$ 220445.30 to US$ 220408 in comparison to method-1. The yearly benefit of US$ 99914.11(20.04%) is achieved with method-2 which is higher than that obtained with method-1.

Although, the results obtained for both the methods are comparable in terms of losses, but the kVArs placed by the two methods makes the difference. Table 2 shows that for the given system with method-1 15,450 kVArs are placed, whereas with method-2 only 14,100 kVArs are placed. This indicates that for OCP problem, NM-PSO provides more efficient solution than PSO.

Table 2. Capacitors Placed at the Candidate Buses

Bus	11	12	21	48	49	50	59	61	64	Total
Method 1	2400	2700	900	3450	600	1200	450	3000	750	15,450
Method 2	2100	2400	900	2400	750	1200	750	2700	900	14,100

Table 3. Capacitors Placed at the Candidate Buses

	Original Configuration	Method 1	Method 2
Peak Power Loss (kW)	1149	809.11	809.07
Medium Power Loss (kW)	563.34	397.51	397.47
Low Power Loss (kW)	134.72	177.81	177.5
Total Power Loss (kW)	1847.06	1384.43	1384.04
% Peak Power Loss Reduction	-	29.58	29.58
% Loss Reduction	-	25.05	25.07
Cost of Peak Power Loss (US$)	193032	135930.50	135923.80
Cost of Energy Loss (US$)	305513.9	220445.30	220408
Capacitor Cost (US$)	-	46350	42300
Total Cost (US$)	498545.9	402725.70	398631.80
Benefits (US$)	-	95820.17	99914.11
% Benefits	-	19.22	20.04

6 Conclusion

In this paper, the proposed NM-PSO based algorithm is tested on IEEE 69-bus radial distribution system for OCP. To make the problem more realistic, objectives of the problem were modified to include the reduction in total cost and real power losses, with inclusion of economic factor and maintenance cost. The work demonstrates that the inclusion of variation in the costs calculated in the objective function gives the problem a more realistic approach. The computational effort for NM-PSO is balanced by dividing the total number population size (3N+1) into NM (N+ 1 particle) and PSO (2N particles). NM being inherently fast in computes the local solutions while the global solutions are searched by the PSO. The results presented help in establishing the NM-PSO as a promising and viable tool for finding the optimal solution and handling more complex, nonlinear objective functions.

References

1. Ng, H.N., Salama, M.M.A., Chikani, A.Y.: Classification of Capacitor Allocation Techniques. IEEE Trans. Power Del. 15(1), 387–392 (2000)
2. Baran, M.E., Wu, F.F.: Optimal Capacitor Placement on Radial Distribution Systems. IEEE Trans. Power Del. 4(1), 725–734 (1989)
3. Baldick, R., Wu, F.F.: Efficient Integer Optimization Algorithms for Optimal Coordination of Capacitors and Regulators. IEEE Trans. Power Syst. 5(3), 805–812 (1990)
4. Abdel-Salam, T.S., Chikhani, A.Y., Hackam, R.: A New Technique for Loss Reduction Using Compensating Capacitors Applied to Distribution Systems with Varying Load Condition. IEEE Trans. Power Del. 9(2), 819–827 (1994)
5. Sundhararajan, S., Pahwa, A.: Optimal Selection of Capacitors for Radial Distribution Systems using a Genetic Algorithm. IEEE Trans. Power Syst. 9(3), 1499–1507 (1994)
6. Salama, M.M.A., Chikhani, A.Y.: An Expert System for Reactive Power Control of a Distribution System Part 1: System Configuration. IEEE Trans. Power Del. 7(2), 940–945 (1992)

7. Ng, H.N., Salama, M.M.A., Chikhani, A.Y.: Capacitor Allocation by Approximate Reasoning: Fuzzy Capacitor Placement. IEEE Trans. Power Del. 15(1), 393–398 (2000)
8. Chang, C.F.: Reconfiguration and Capacitor Placement for Loss Reduction of Distribution Systems by Ant Colony Search Algorithm. IEEE Trans. Power Syst. 23(4), 1747–1755 (2008)
9. Mekhamer, S.F., Soliman, S.A., Moustafa, M.A., El-Hawary, M.E.: Application of Fuzzy Logic for Reactive Power Compensation of Radial Distribution Feeders. IEEE Trans. Power Syst. 18(1), 206–213 (2003)
10. Fan, S.S., Zahara, E.: A Hybrid Simplex Search and Particle Swarm Optimization for Unconstrained Optimization. Eur. Jr. Operational Research 181, 527–548 (2007)
11. Zahara, E., Kao, Y.: Hybrid Nelder-Mead Simplex Search and Particle Swarm Optimization for Constrained Engineering Design Problems. Expert Syst. with App. 36, 3880–3886 (2009)
12. da Silva, I.C., Carneiro, S., de Oliveira, E.J., de Souza Costa, J., Pereira, J.L.R., Garcia, P.A.N.: A Heuristic Constructive Algorithm for Capacitor Placement on Distribution Systems. IEEE Trans. Power Syst. 23(4), 1619–1626 (2008)
13. Abido, A.: Optimal Power Flow Using Particle Swarm optimization. Int. Jr. Elect. Power Energy Syst. 24(7), 563–571 (2002)
14. Abido, A.: Particle Swarm Optimization for Multimachine power System Stabilizer Design. Power Engg. Scoiety Summer Meeting (PES) 3, 1346–1351 (2001)
15. Nelder, J.A., Mead, R.: A Simplex Method for Function Minimization. Comp. Jr. 7, 308–313 (1965)
16. Cheng, C.S., Shiromohammadi, D.: A Three-phase power Flow Method for Real-Time Distribution System Analysis. IEEE Trans. Power Syst. 10(2), 671–679 (1995)
17. Huang, S.-J.: An Immune Based Optimization Method to Capacitor Placement in a Radial Distribution System. IEEE Trans. Power Del. 15(2), 744–749 (2000)

Comparative Performance Study of Genetic Algorithm and Particle Swarm Optimization Applied on Off-grid Renewable Hybrid Energy System

Bhimsen Tudu, Sibsankar Majumder,
Kamal K. Mandal, and Niladri Chakraborty

Power Engineering Department,
Jadavpur University, Kolkata: 700098, India
`bhimsen_ju@yahoo.co.in, Shivshankar.Mazumder@gmail.com,`
`kkm567@yahoo.co.in, chakraborty_niladri@hotmail.com`

Abstract. This paper focuses on unit sizing of stand-alone hybrid energy system using Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) and comparative performance study of these two meta-heuristic techniques on hybrid energy system. The hybrid system is designed focusing on the viability and combining different renewable energy sources like wind turbines, solar panels along with micro-hydro plant as well as fuel cells to compensate the deficit generation in different hours. Apart from the non-conventional sources, the system has been optimized with converters, electrolyzers and hydrogen tanks. Net present cost (NPC), cost of energy (COE) and generation cost (GC) for power generation have been considered while optimal unit sizing of the system are performed. Feasibility of the system is made based on net present cost (NPC). The performances of two algorithms have been checked for different values of variants of the respective algorithms and a comparative study has been carried out based on number of iterations taken to find optimal solution, CPU utilization time and also quality of solutions. The comparative analysis shows that the Particle Swarm Optimization technique performs better than Genetic Algorithm when applied for the sizing problem.

Keywords: Genetic Algorithm, Particle Swarm Optimization, Off-grid renewable energy system, Solar panels, Wind turbine, Micro-hydro plant.

1 Introduction

Fast depletion of fossils fuels, direct and indirect green house gases emission from thermal power plants and environmental issues have shifted world's focus towards the new and renewable sources of energy. Huge amount of electricity is being produced to meet the high energy demand in India and is in fifth position in electricity generation (922.2TWh) in the world at the end of year 2010 [1]. Still many remote

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 151–158, 2011.
© Springer-Verlag Berlin Heidelberg 2011

villages of India are waiting for the electricity to taste. It is now of great concern for government to provide power to a remote location where power can't be supplied from the main grid line due to the high associated cost. This has explored a new horizon and opened up the door of off-grid renewable energy system. For better understanding of performance and proper optimization of this type of hybrid system, different meta-heuristic techniques have been used extensively such as Genetic Algorithm [2-4], Particle Swarm Optimization [5-7], Fuzzy Control, Differential Evolutionary Algorithm, Ant Colony, Tabu Search etc. [8-11].

2 Hybrid Test System Model

The hybrid test system model has been framed keeping in mind a typical village of India. It is assumed that the village has high potential of solar energy, wind energy along with the micro hydro plant. The sizing of the proposed hybrid energy system has been carried out considering an increase of 10% in average load profile over the lifetime of the project of 30 years. The average hourly wind velocity and solar insolation data and load profile for 24 hours of that location are shown in Figure 1.

The different components used in this system are standard components of different manufacturer make and data used for different components are listed in Table 1.

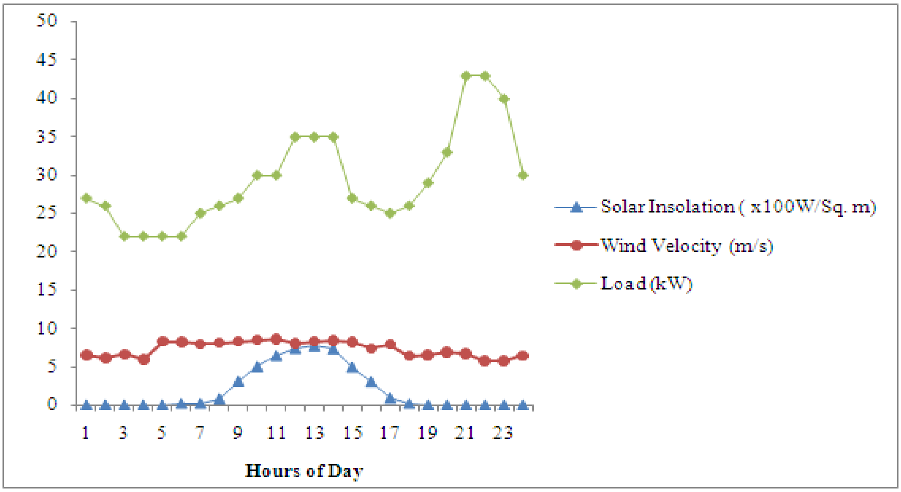


Fig. 1. Hourly load, solar insolation and wind velocity data for the site

Table 1. Different parameter values & costs (Rs./Yr) considered for the hybrid energy system

Component	Parameter	Value	Capital Cost	Replacement Cost	Operation Cost
Micro Hydro-plant	Installed capacity	20kW	1610K	1288K	32K
	Operating efficiency	50%			
	lifetime	20Years			
Wind Turbine	Capacity	10kW	1334K	1104K	46K
	Rated speed	12m/s			
	Cut–in speed	2.5m/s			
	Furling speed	15.6m/s			
	Lifetime	15years			
Solar PV Module	Rated power	0.15kW	23K	18.4K	NIL
	Slope	25 degrees			
	Derating Factor	0.9			
	Lifetime	20 years			
Fuel Cell	Capacity	5kW	552K	460K	23K
	Efficiency	50%			
	Lifetime	15Years			
Electrolyzer	Capacity	1kW	92K	82K	9.2K
	Efficiency	90%			
	Lifetime	20 Years			
Converter	Capacity	5kW	115K	92K	0.92K
	Efficiency	90%			
	Lifetime	20 Years			
Hydrogen Tank	Capacity	1 Kg	55.2K	46K	0.46K
	Lifetime	15 Years			

3 System Costs and Objective Function

The total annualized cost of a component consists of the annualized capital cost, annualized replacement cost, annualized operation and maintenance cost as well as annualized fuel cost. The total annualized cost of different components of the hybrid systems can be represented by the equations shown below [12].

Total annualized cost of each component:

$$C_{ann,\,component} = C_{acap,\,component} + C_{arep,\,component} + C_{aom,\,component} + C_{afc,\,component} \tag{1}$$

Where, C_{acap} = annualized capital cost, C_{arep} = annualized replacement cost, C_{aom} = annualized operation & maintenance cost, C_{afc} = annualized fuel cost.

The total annualized cost of the hybrid energy system can be expressed as [12]:

$$\begin{aligned} C_{ann,\ tot} &= \sum_{s=1}^{N_s} C_{ann,\ solar,\ s} + \sum_{w=1}^{N_w} C_{ann,\ wind,\ w} + \sum_{c=1}^{N_c} C_{ann,\ converter,\ c} \\ &+ \sum_{t=1}^{N_t} C_{ann,\ \tan k,\ t} + \sum_{e=1}^{N_e} C_{ann,\ electo,\ e} + \sum_{f=1}^{N_f} C_{ann,\ fuelcell,\ f} + C_{ann,\ hydro} \end{aligned} \tag{2}$$

Where, N_s = number of installed solar modules, N_w = number of installed wind turbines, N_c = number of installed converters, N_t = number of installed hydrogen tanks, N_e = number of installed electrolyzers and N_f = number of fuel cells installed in the hybrid system.

The net present cost of the system is expressed as:

$$C_{NPC} = \frac{C_{ann,\ tot}}{CRF_{proj}} \tag{3}$$

The cost of energy of the system is given by [13] as:

$$COE = \frac{C_{ann,\ tot}}{Total\ \ load\ \ catered\ \ by\ \ the\ \ system\ \ over\ \ the\ \ year\ \ in\ \ kWh} \tag{4}$$

Similarly the generation cost of the system can be expressed as:

$$GC = \frac{C_{ann,\ tot}}{Total\ \ power\ \ generated\ \ over\ \ the\ \ year\ \ in\ \ kWh} \tag{5}$$

As in some hours of operation, the total power generated by the system is not completely utilized by the demand side and there is some input to the electrolysers, the value of COE and GC differs by some extent. In this study, the system is optimized based on the net present cost.

4 Optimization Methods

The objective function derived above for obtaining the optimal sizing of the different components of the hybrid energy system under study has been optimized employing Particle Swarm Optimization technique and Genetic Algorithm.

4.1 Particle Swarm Optimization

Particle swarm optimization was originally developed by Kenedy and Eberhart [14] based on natural behaviour of the particles or agents and it is quite efficient population based optimization procedure for multidimensional non-linear complex problem. Each position of the particle in d-dimensional search space represents the possible solution of the problem. After each iteration, particles change their position to give the better solution. Each time particles update their position in the search space depending upon their own experience and their neighbour particles' experience. A particle's position is updated depending upon its current velocity, its own distance from its previous best position and distance from the best position occupied by the particle in the group. According to Kennedy et al., for each particle *i,* its position is updated in the following manner:

$$X_{k+1}^{i} = X_{k}^{i} + V_{k+1}^{i} \tag{6}$$

Where V_{k+1}^{i} is pseudo velocity and can be calculated as follows:

$$V_{k+1}^{i} = w_k V_k^i + c_1 r_1 (P_k^i - X_k^i) + c_2 r_2 (P_k^g - X_k^i) \tag{7}$$

Here k represents a pseudo time increment, P_k^i represents the best ever position of particle i at time k and P_k^g represents the global best position in the group of swarm. r_1 & r_2 represent the uniform random values in the range [0 1], w represents the weight parameters and c_1 & c_2 are the acceleration constant which depend on the cognitive and social nature of the swarms.

4.2 Genetic Algorithm

Genetic algorithm is started with a random initial population of chromosomes. The fitness function is evaluated for each chromosome of the corresponding population. A selection operation is performed on the population of chromosomes which will be subjected to crossover and mutation operations based on Roulette Wheel method [2] to produce the new generation. The minimum value of objective function found for a generation of chromosomes is compared with the minimal value found in the previous generation. If the current minimal value is lower than that of the previous generation, then this value is considered to be the optimal solution of the minimization problem. The optimal chromosome found in each generation represents the sizing of different components for the hybrid energy system.

From one generation to next generation four steps are involved:

a) **Selection:** selection of individuals (chromosomes) for reproduction according to their fitness (objective function value)
b) **Crossover:** merging the genetic information of two individuals based on crossover probability factor.
c) **Mutation:** a random alternation of the chromosomes based on mutation probability factor. The positive effect is preservation of genetic diversity and as an effect that local maxima can be avoided.
d) **Sampling:** procedures which computes a new generation from the previous one and its off-spring.

5 Simulated Results and Comparative Study

The objective function formulated for obtaining the sizing of the hybrid energy system is optimized using both Particle Swarm Optimization and Genetic Algorithm.

A comparative analysis is performed varying the values of different optimization variants to find their best fitted values for the objective function under study.

The optimal sizing as found employing Genetic Algorithm and Particle Swarm Optimization is given in Table 2.

We performed the analysis taking the different values of acceleration constant and swarm population for Particle Swarm Optimization technique keeping weight parameter constant. We considered different values for acceleration constant in the range [0.1 2]. Similarly for swarm population, different values in the range [10 200] were considered. Based on our analysis, we found that the technique performs best when acceleration constant is 2 and the swarm population is 150.

Table 2. Optimal sizing obtained for the integrated hydro-wind-solar-fuel cell system

Component	Optimal Sizing (No. of Units)	Capacity
Wind Turbine	8	80 kW
Solar Panel	0	0 kW
Converter	3	15 kW
Fuel Cell	3	15 kW
Electrolyser	5	5 kW
Hydrogen Tank	3	3 kg.
Net Present Cost (NPC): Rs. 24327.45k		
Generation Cost (GC): Rs. 4.98/kWh		
Cost of Energy (COE): Rs. 7.53/kWh		

Again, we performed similar analysis for Genetic Algorithm varying the values of crossover probability factor, mutation probability factor and chromosome population. We considered different values for crossover and mutation probability factors in the range [0.1 0.9]. In the same way, for chromosome population, different values in the range [10 200] were considered. Analyzing the performance of Genetic Algorithm in different runs, it was found that it performs best for the objective function when crossover and mutation probabilities are 0.5 and chromosome population is 200.

Using the fine tuned values obtained by varying the parameters of PSO and GA, the comparative analysis between the two optimization techniques was performed. The comparison of performance was mainly based on the number of iterations needed to reach the optimal value, CPU utilization time as well as quality of solutions.

A sampling of 100 separate runs was taken for each of the techniques. The performances of these two optimization techniques in terms of average number of iteration needed to find the optimal value and the average CPU Utilization time are given in Table 3.

Table 3. Performance comparison of PSO and GA

Method	Avg. No. Of Iteration to Find the Optimal Value	CPU Utilization Time (Sec.)
PSO	2	12.7
GA	13	108.5

Figure 2 shows the number of iteration to reach the optimal value with the fine tuned values of the variants for Genetic Algorithm and Particle Swarm Optimization.

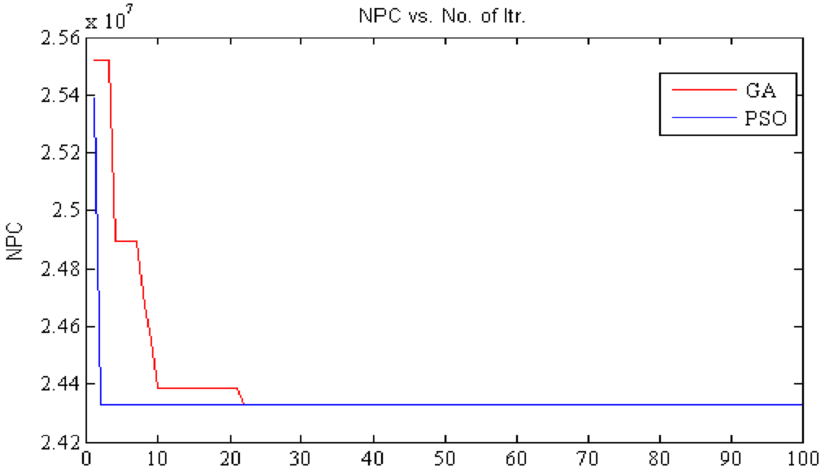


Fig. 2. The optimal Net Present Cost (NPC) in terms of the iterations for GA & PSO

5 Conclusion

From the analysis, it can be derived that PSO performs a way ahead of Genetic Algorithm both in terms of minimum number of iterations needed to reach the optimum as well as CPU utilization time period. When analysis was performed for samples of 100 different runs, we found that PSO reaches the optimal value using 85% less iterations and consuming 88% less CPU time. So PSO is a more preferred technique of optimization rather than Genetic Algorithm for our test system.

References

1. BP Statistical Review of World Energy (June 2011), `http://www.bp.com`
2. Koutroulis, E., Kolokotsa, D., Potirakis, A., Kostas, K.: Methodology for optimal sizing of stand-alone photovoltaic/wind-generator systems using genetic algorithms. Solar Energy 80, 1072–1088 (2006)
3. Dufo-Lopez, R., Bernal-Agustin, J.L.: Design and control strategies of PV-Diesel systems using genetic algorithms. Solar Energy 79, 33–46 (2005)

4. Ould Bilal, B., Sambou, V., Ndiaye, P.A., Kebe, C.M.F., Ndongo, M.: Optimal design of a hybrid solar-wind-battery system using the minimization of the annualized cost system and minimization of the loss of power supply probability (LPSP). Technical note, Renewable Energy 35, 2388–2390 (2010)
5. Hakimi, S.M., Moghaddas-Tafreshi, S.M.: Optimal sizing of a stand-alone hybrid power system via particle swarm optimization for Kahnouj area in south-east of Iran. Renewable Energy 34, 1855–1862 (2009)
6. Hakimi, S.M., Tafreshi, S.M., Kashefi, A.: Unit sizing of a stand-alone hybrid power system using particle swarm optimization (PSO). In: Proceedings of the IEEE International Conference on Automation and Logistics, Jinan, China, August 18-21 (2007)
7. Hakimi, S.M., Tafreshi, S.M., Rajati, M.R.: Unit sizing of a stand-alone hybrid power system using model free optimization. In: 2007 IEEE International Conference on Granular Computing (2007)
8. Li, M., Wang, C.: Research on optimization of wind and PV hybrid power systems. In: Proceedings of the World Congress on Intelligent Control and Automation, Chongqing, China, June 25-27 (2008)
9. Qi, Y., Jianhua, Z., Zifa, L., Shu, X., Weiguo, L.: A new methodology for optimizing the size of hybrid PV/wind system. In: ICSET (2008)
10. Xu, D., Kang, L., Cao, B.: Graph-Based Ant System for Optimal Sizing of Standalone Hybrid Wind/PV Power Systems. In: Huang, D.-S., Li, K., Irwin, G.W. (eds.) ICIC 2006. LNCS (LNAI), vol. 4114, pp. 1136–1146. Springer, Heidelberg (2006)
11. Katsigiannis, Y.A., Georgilakis, P.S.: Optimal sizing of small isolated hybrid power systems using tabu search. Journal of Optoelectronics and Advanced Materials 10(5), 1241–1245 (2008)
12. Dufo-Lopez, R., Bernal-Agustin, J.L.: Design and control strategies of PV-Diesel systems using genetic algorithms. Solar Energy 79, 33–46 (2005)
13. Katsigiannis, Y.A., Georgilakis, P.S., Karapidakis, E.S.: Genetic Algorithm Solution to Optimal Sizing Problem of Small Autonomous Hybrid Power Systems. In: Konstantopoulos, S., Perantonis, S., Karkaletsis, V., Spyropoulos, C.D., Vouros, G. (eds.) SETN 2010. LNCS, vol. 6040, pp. 327–332. Springer, Heidelberg (2010)
14. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks IV, pp. 1942–1948 (1995)

An Efficient Algorithm for Multi-focus Image Fusion Using PSO-ICA

Sanjay Agrawal, Rutuparna Panda, and Lingaraj Dora

Department of Electronics & Telecommunication Engineering,
VSS University of Technology, Burla, Odisha, India
agrawals_72@yahoo.com

Abstract. A pixel-level multi-focus image fusion scheme using Independent Component Analysis (ICA) and Particle Swarm Optimization (PSO) is proposed in this paper. The novelty in this work is optimization of ICA bases using PSO and its application to multi-focus image fusion, which is not found in the literature. The idea is to divide the input registered images into patches and get the independent components using ICA transform. The images are then fused in the transform domain using pixel-based fusion rules. PSO is used to optimize the independent components in ICA. We observe that the proposed method outperforms the existing fusion techniques using ICA.

1 Introduction

The process of combining multiple images of the same object into a single image is known as Image Fusion. A wide variety of data acquisition devices are available at present. Hence, the scope of image fusion has increased manifold and it has become an important area of research. There are now sensors available which cannot generate images of all objects at various distances (from the sensor) with equal clarity (e.g. camera with finite depth of field, light optical microscope and so on). Thus several images of an object are captured, with focus on different parts of it. The captured images are complementary in many ways and a single one of them is not sufficient in terms of their respective information content. The advantage of multi-focus data can be fully exploited by integrating the sharply focused regions seen in the different images using suitable image fusion techniques [1-3]. Multi-focus image fusion techniques have been widely used in the field of image analysis tasks such as target recognition, remote sensing and medical imaging.

A number of multi-focus image fusion techniques are seen in the literature. The techniques in which the fusion operation is performed directly on the source images (e.g. weighted- average method) often have serious side effects like reduction of visual perception of the fused image. Other approaches include, image fusion using controllable camera [4], probabilistic methods [5], image gradient method with majority filtering [6], multi-scale methods [7] and multi-resolution methods [8, 9]. Maurer et al. [10] described methods depending on controlled camera motion but that do not work for arbitrary set of images. Probabilistic techniques involve huge computation using floating point arithmetic and thus require a lot of time and memory-space. Image gradient method with majority filtering has the drawback that the defocused

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 159–166, 2011.
© Springer-Verlag Berlin Heidelberg 2011

zone of one image is enhanced at the expense of focused zone of others. Multi-scale methods involve processing and storing of scaled data at various levels which are of same size as that of the original images resulting in a huge amount of memory and time requirement.

In this paper, the use of PSO in optimizing the independent components in ICA for a pixel-based multi-focus image fusion algorithm is introduced. The idea is to divide the source images into patches and then express the patches as a linear combination of a set of basis images. It is assumed that the input images have negligible registration problem. Then an analysis kernel and a synthesis kernel is estimated. The transform projects the observed signal on a set of basis vectors. The estimation of these vectors is performed using a population of training image patches and a criterion (cost function), which is going to be optimized using PSO in order to select the basis vectors. First PCA is used to identify the uncorrelated components and then statistically independent vectors are identified using ICA. It is here only PSO is used to obtain and optimize the independent bases which have been identified by using FastICA method [12]. The advantage of our method is in using PSO to optimize the independent components than using a random approach. And the resultant fused images are both qualitatively and visually superior to that of FastICA.

The rest of this paper is organized as follows. Section 2 gives the introduction of Independent Component Analysis (ICA). Section 3 introduces the framework of PSO. The proposed fusion scheme using PSO-ICA is described in Section 4. Section 5 gives simulation results and Section 6 gives the conclusion.

2 Independent Component Analysis

The image fusion process can be performed at different levels i.e. signal, pixel, feature, and symbolic level of information representation. Nikolov *et al.* [15] proposed a classification of image fusion algorithms into spatial domain and transform domain techniques. The transform domain image fusion scheme consists of obtaining a transform on each input image and, following specific fusion rules, combining them into a composite transform domain representation. The composite output image is obtained by applying the inverse transform on this composite transform domain representation. Instead of using a standard bases system, such as the DFT, the mother wavelet or cosine bases of the DCT, a set of bases that are suitable for a specific type of image can be trained. A training set of image patches, which are acquired randomly from images of similar content, can be used to train a set of statistically independent bases. This is known as independent component analysis (ICA) [12].

In order to obtain a set of statistically independent bases for image fusion in the ICA domain, training is performed with predefined set of images. Training images are selected in such a way that the content and statistical properties are similar for the training images and the images to be fused. An input image $I(x,y)$ is randomly windowed using a rectangular window W of size NxN, centered around the pixel (m_o,n_o). The result of windowing is an image patch which is defined as [14]

$$I_p(m,n) = W(m,n) * I\,(m_0 - N/2 + m,\ n_0 - N/2 + n)\ . \qquad (1)$$

where m and n take integer values from the interval [0,N-1].

Each image patch $I_p(m,n)$ can be represented by a linear combination of a set of M basis patches $b_i(m,n)$,

$$I_p(m,n) = \sum_{i=1}^{M} v_i b_i(m,n) \quad . \tag{2}$$

where $v_1, v_2, \ldots v_M$ stands for the projections of the original image patch on the basis patch i.e. $v_i = \left\langle I_p(m,n), b_i(m,n) \right\rangle$. 2-D representation of the image patches can be simplified to a 1-D representation, using lexicographic ordering. This implies that an image patch I_p (m, n) is reshaped into a vector $\underline{I}_p$, mapping all the elements from the image patch matrix to the vector in a row-wise or column-wise fashion. Decomposition of image patches into a linear combination of basis patches can then be expressed as follows:

$$\underline{I}_p(t) = \sum_{i=1}^{M} v_i(t)\underline{b}_i = [\underline{b}_1, \underline{b}_2 \ldots\ldots\ldots\ldots \underline{b}_M] * \begin{bmatrix} v_1(t) \\ v_2(t) \\ \ldots\ldots\ldots \\ v_M(t) \end{bmatrix} . \tag{3}$$

where t represents the image patch index.

Let $B = [\underline{b}_1, \underline{b}_2 \ldots\ldots\ldots\ldots \underline{b}_M]$ and $\underline{V}(t) = [v_1, v_2 \ldots\ldots\ldots\ldots v_M]^T$ then equation (3) can be represented as

$$\underline{I}_p(t) = B * \underline{V}(t). \tag{4}$$

$$\underline{V}(t) = B^{-1} * \underline{I}_p(t) = A * \underline{I}_p(t) . \tag{5}$$

Thus, here $A = [\, a_1, a_2 \ldots\ldots a_M \,]^T$ represents an unknown mixing matrix called analysis kernel and B represents the un-mixing matrix called synthesis kernel. This “transform” projects the observed signal $\underline{I}_p$ (t) on a set of basis vectors $\underline{b}_i$. So in the first stage, PCA is used for dimension reduction. This is obtained by eigen value decomposition of the data correlation matrix $C = E\ \{\underline{I}_p(t)\underline{I}_p(t)^T\}$. The eigen values of the correlation matrix illustrate the significance of their corresponding basis image patches [13]. If V is the obtained $K \times N^2$ PCA matrix, the input image patches are transformed by

$$\underline{Z}(t) = V * \underline{I}_p(t). \tag{6}$$

After the PCA preprocessing step, the statistically independent vectors are selected by using approximations of negentropy by minimizing the objective function

$$J_G(w) = [E\ \{G\ (w^T z)\} - E\ \{G\ (v)\}]^2 \quad . \tag{7}$$

where w is a m-dimensional (weight) vector constrained so that $E\{(w^T x)^2\} = 1$,

E{.} represents the expectation operator, ‘v’ is a Gaussian variable with zero mean and unit variance and G(.) is a non-quadratic function.

The above function computes one independent component. But for several independent components the following optimization problem is solved using PSO.

$$\text{minimize } \sum_{i=1}^{n} J_G(w_i) \text{ wrt. } w_i,\ i=1,\ldots,n\,. \tag{8}$$

under constraint $E\{(w_k^T x)(w_j^T x\} = \delta_{jk}$.

After the input image patches $\underline{I}_p(t)$ are transformed to their ICA domain representations $\underline{V}(t)$, image fusion can be performed in the ICA domain. The equivalent vectors $V_k(t)$ from each image are combined in the ICA domain to obtain a new image $V_f(t)$ as

$$V_f(t) = g(V_1(t),\ldots,V_k(t)). \tag{9}$$

where g(*) represents the fusion rule. Many fusion rules have been proposed earlier and for multi-focus image fusion, "max-abs" rule i.e. fusion by absolute maximum by selecting the greatest in absolute value of the corresponding coefficients in each image is chosen as

$$V_f(t) = \mathrm{sgn}(V_k(t)) \max_k |V_k(t)|. \tag{20}$$

After the composite image $V_f(t)$ is constructed in the ICA domain, the spatial domain representation can be obtained by using the synthesis kernel B, and synthesize the image

$$I_f(x,y) = B^* V_f(t). \tag{11}$$

3 Particle Swarm Optimization

The particle swarm optimization is an evolutionary computation technique which is a population-based optimization tool that was proposed by Eberhart and Kennedy [16] in 1995.

Let N denote the number of particles in a swarm. In general, there are three attributes of a particle, the particle's current position x_i, current velocity v_i and local best position p_i, in the search space to present their features. Each particle in the swarm is iteratively updated according to the aforementioned attributes. During the search process, the particle successively adjusts its position towards the global optimum according to the two factors: the best position encountered by itself (*pbest*) which is denoted as $p_{i,j} = (p_{i1}, p_{i2}, \ldots, p_{iD})$ and the best position encountered by the whole swarm (*gbest*) which is denoted as $p_g = (p_{g1}, p_{g2}, \ldots, p_{gD})$.

Assuming that the objective function *f is* to be minimized (or maximized), the new velocity of every particle is updated as given below:

$$\begin{aligned} v_{i,j}(g+1) &= w^* v_{i,j}(g) + c_1{}^*\mathrm{rand1}(g)^*[\mathrm{pbest}_{i,j}(g) - x_{i,j}(g)] \\ &\quad + c_2{}^*\mathrm{rand2}(g)^*[\mathrm{gbest}_{i,j}(g) - x_{i,j}(g)] \text{ for all } j \in 1\ldots N. \end{aligned} \tag{12}$$

$v_{i,j}$ is the velocity of the jth dimension of the ith particle, w denotes the inertia weight of velocity, c_1 and c_2 denotes acceleration coefficients namely cognitive and social parameter respectively, rand1 and rand2 represent two uniform random functions in the range (0,1) to effect the stochastic nature of the algorithm and g is the number of generations.

The new position of a particle is calculated as

$$x_{i,j}(g+1) = x_{i,j}(g) + v_{i,j}(g+1). \tag{13}$$

4 Proposed PSO-ICA Algorithm for Multi-focus Image Fusion

The algorithm of the proposed image fusion method using PSO-ICA is as follows. A set of images with similar content is chosen as the input images. We assume the input images have negligible registration problem. The steps followed in the proposed method are:

Step 1 decomposes the input registered images into non overlapping square patches using equation (1). The acquired patches are then ordered lexicographically using equation (3).

Step 2 applies the PSO-ICA transform on individual input image patches and converts the images from spatial domain to transform domain using equation (6). We have used the optimization of negentropy as a non- Gaussianity measurement to identify the independent components as explained in [13].

Step 3 uses the max-abs fusion rule by combining the corresponding coefficients from each transformed image to get the fused image using equation (10).

Step 4 uses the inverse transformation to convert the fused image into the spatial domain using equation (5).

PSO has been used in this paper to minimize the objective function as given in equation (8).

The steps followed for PSO-ICA are as follows:

- The parameters for PSO are initialized including number of particles in the swarm N, acceleration coefficients c1, c2, inertia weight w and the maximum iterative count.
- X, v, pbest are initialized for each particle and gbest for the swarm.
- The linear matrix w is calculated for each particle using equation (9).
- The fitness value is then calculated for each particle.
- Pbest and Gbest is calculated for each particle and the swarm respectively.
- The velocity and the position for each particle is updated using equation (12) and (13)

The termination condition in the proposed method is the number of iterations or no improvement in ‘gbest’ in a number of iterations.

5 Simulation Results

In order to demonstrate the effectiveness of the proposed method, three artificially distorted aircraft images are taken as presented in fig.1. Because a reference image is available, numerical evaluation of the fusion schemes is presented in Table 1.

The Signal to Noise Ratio (SNR) expression to compare the reference image with the fused image is:

$$\mathrm{SNR}_{(\mathrm{dB})} = 10\log_{10} \frac{\sum_m \sum_n I_{ref}(m,n)^2}{\sum_m \sum_n (I_{ref}(m,n) - I_f(m,n))^2} . \tag{14}$$

The expression for Image Quality Index (Q_0) as a performance measure is :

$$Q_0 = \frac{4\sigma_{I_{ref}I_f} m_{I_{ref}} m_{I_f}}{(m_{I_{ref}}^2 + m_{I_f}^2)(\sigma_{I_{ref}}^2 + \sigma_{I_f}^2)} . \tag{15}$$

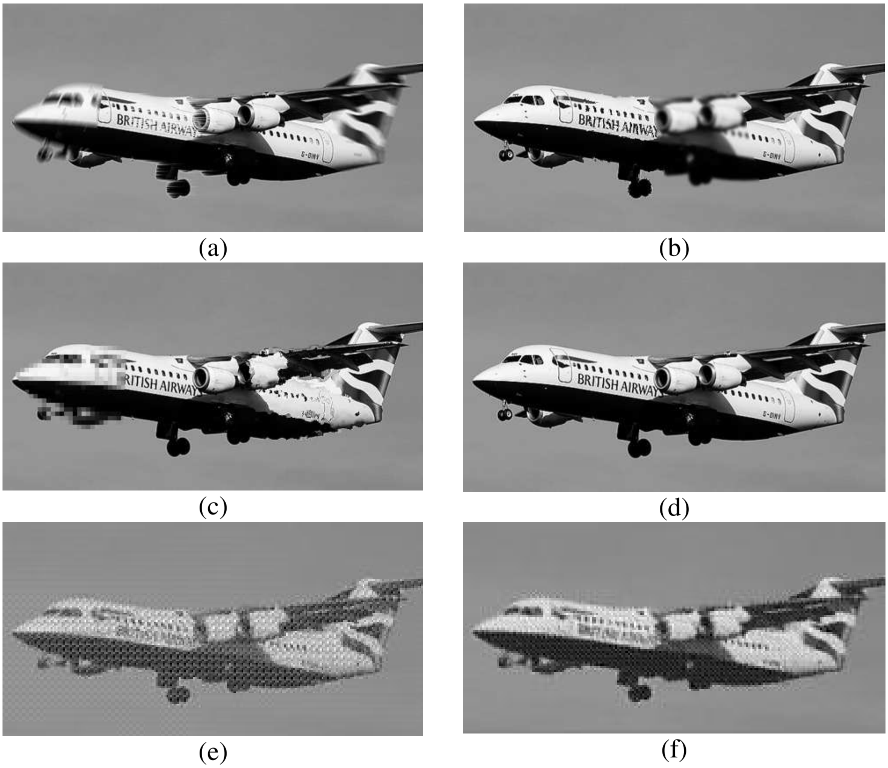
where

$$\sigma_I^2 = \frac{1}{M_1 M_2 - 1} \sum_{m=1}^{M_1} \sum_{n=1}^{M_2} (I(m,n) - m_I)^2$$

$$\sigma_{IJ} = \frac{1}{M_1 M_2 - 1} \sum_{m=1}^{M_1} \sum_{n=1}^{M_2} (I(m,n) - m_I)(J(m,n) - m_J)$$

m_I represents mean of image I(m,n), M_1, M_2 represent size of the images. As $-1 \leq Q_0 \leq 1$, a value of Q_0 that is closer to 1 indicates better fusion performance.

$$RMSE = \frac{\sqrt{\sum \sum [\mathrm{I_{ref}} - \mathrm{I_f}]^2}}{\sqrt{\mathrm{M_1 M_2}}} \tag{16}$$



(a) (b)

(c) (d)

(e) (f)

Fig. 1. Artificially Distorted Aircraft Source Images and Fusion Results: (a) Aircraft 1,(b) Aircraft 2, (c) Aircraft 3, (d) Reference Image, (e) Fast-ICA Fusion Result, (f) PSO-ICA Fusion Result.

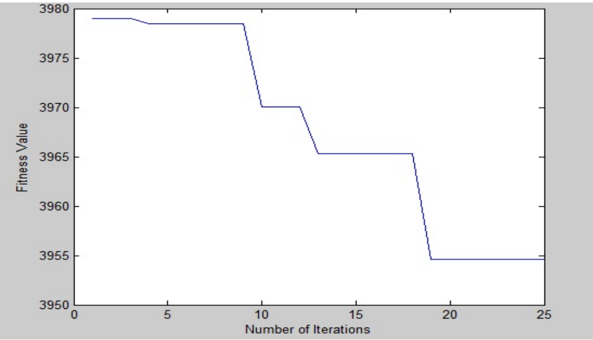


Fig. 2. Fitness function value vs No. of Iterations

Table 1. Performance Comparison of Fast-ICA and PSO-ICA fusion method

Fusion Parameters	Fast-ICA	PSO-ICA
Signal to Noise Ratio (SNR in DB)	13.5384	17.6621
Image Quality Index (Qo)	0.7881	0.9295
Root Mean Square Error (RMSE)	0.0968	0.1171
Information Entropy (E)	5.8284	6.0638
Signal to Noise Ratio (SNR in DB)	13.5384	17.6621

We trained the ICA bases with 8 x 8 non overlapping image patches from the input images. Then we implemented the fusion method using Fast-ICA and PSO ICA. We can see that we got better fusion performance using PSO-ICA both in visual quality and metric quality (PSNR, Q_0). The reason may be, in Fast-ICA to find out the independent components no standard optimization method is used but we have introduced PSO in ICA to estimate the independent components. The parameters for PSO are taken as:

Swarm size = 40, Dimension = size of W, w = 0.9 to 0.4, $c_1 = c_2 = 2$, Number of Iterations = 25.

6 Conclusion

This paper has demonstrated that the proposed method outperforms the existing Fast-ICA technique both visually and quantitatively. Though PSO and ICA are not new but their application to multi-focus image fusion is new. The problem is not multimodal, that's why the standard PSO has been taken. The results are quite evident by the quantitative parameters as given in the table. Future work should include some more example figures to prove that proposed method is better. Also some more comparisons with other swarming methods can be included in future.

References

1. De, I., Chanda, B.: A simple and efficient algorithm for multifocus image fusion using morphological wavelets. Signal Processing 86, 924–936 (2006)
2. Zaveri, T., Zaveri, M.: A Novel Two Step Region Based Multifocus Image Fusion Method. International Journal of Computer and Electrical Engineering 2(1) (February 2010)
3. van Leeuwen, J. (ed.): Computer Science Today. Recent Trends and Developments. LNCS, vol. 1000. Springer, Heidelberg (1995)
4. Seales, W., Dutta, S.: Everywhere-in-focus image fusion using controllable cameras. In: Proceedings of SPIE, vol. 2905, pp. 227–234 (1996)
5. Bloch, I.: Information combination operators for data fusion: a review with classification. IEEE Trans. SMC: Part A 26, 52–67 (1996)
6. Eltoukhy, H.A., Kavusi, S.: A computationally efficient algorithm for multi-focus image reconstruction. In: Proceedings of SPIE Electronic Imaging (June 2003)
7. Mukhopadhyay, S., Chanda, B.: Fusion of 2d gray scale images using multiscale morphology. Pattern Recognition 34, 1939–1949 (2001)
8. Yang, X., Yang, W., Pei, J.: Different focus points images fusion based on wavelet decompo-sition. In: Proceedings of Third International Conference on Information Fusion, vol. 1, pp. 3–8 (2000)
9. Zhang, Z., Blum, R.S.: Image fusion for a digital camera application. In: Conference Record of the Thirty-Second Asilomar Conference on Signals, Systems and Computers, vol. 1, pp. 603–607 (1998)
10. Maurer, D.E., Baker, J.P.: Fusing multimodal biometrics with quality estimates via a Bayesian belief network. Pattern Recognition 41, 821–832 (2008)
11. Hyvrinen, A., Karhunen, J., Oja, E.: Independent Component Analysis. Wiley, London (2001)
12. Nian, F., Li, W., Sun, X., Li, M.: An Improved Particle Swarm Op-timization Application to Independent Component Analysis. In: ICIECS 2009, pp. 1–4 (2009)
13. Mitianoudis, N., Stathaki, T.: Pixel-based and region-based image fusion using ICA bases. Information Fusion 8(2), 131–142 (2007)
14. Cvejic, N., Bull, D., Canagarajah, N.: Region-Based Multimodal Image Fusion using ICA Bases. IEEE Sensors Journal 7(5) (May 2007)
15. Nikolov, S., Bull, D., Canagarajah, N.: Wavelets for image fusion. In: Petrosian, A., Meyer, F. (eds.) Wavelets in Signal and Image Analysis, Computational Imaging and Vision Series, pp. 213–244. Kluwer Academic Publishers, Dordrecht (2001)
16. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In: Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ, pp. 1942–1948 (1995)
17. Liang, J.J., Qin, A.K., Suganthan, P.N., Baskar, S.: Comprehensive Learning Particle Swarm Optimizer for Global Optimization of Multimodal Functions. IEEE T. on Evolutionary Computation 10(3), 281–295 (2006)
18. Liang, J.J., Suganthan, P.N.: Dynamic Multi-Swarm Particle Swarm Optimizer. In: IEEE Swarm Intelligence Symposium, Pasadena, CA, USA, pp. 124–129 (June 2005)

Economic Emission OPF Using Hybrid GA-Particle Swarm Optimization

J. Preetha Roselyn[1], D. Devaraj[2], and Subranshu Sekhar Dash[3]

[1] SRM University, Kattankulathur-603203
`preetha.roselyn@gmail.com`
[2] Kalasalingam University, Srivilliputhur-626190
`deva230@yahoo.com`
[3] SRM University, Kattankulathur-603203
`munu_dash_2k@yahoo.com`

Abstract. This paper presents a Hybrid Genetic Algorithm (HGA) Particle Swarm Optimization (PSO) approach to solve Economic Emission Optimal Power Flow problem. The proposed approach optimizes two conflicting objective functions namely, fuel cost minimization and emission level minimization of polluted gases namely NO_X, SO_X and CO_x simultaneously while satisfying operational constraints. An improved PSO which permits the control variables to be represented in their natural form is proposed to solve this combinatorial optimization problem. In addition, the incorporation of genetic algorithm operators in PSO improves the effectiveness of the proposed algorithm. The validity and effectiveness have been tested with IEEE 30 bus system and the results show that the proposed algorithm is competent in solving Economic Emission OPF problem in comparison with other existing methods.

1 Introduction

OPTIMAL Power Flow (OPF) is one of the most important function in modern energy management systems. The OPF problem aims to achieve an optimal solution to specific objective function such as fuel cost by adjusting the power system control variables while satisfying a set of operational and physical constraints. A number of mathematical programming based techniques have been proposed to solve the optimal power flow problem. These include gradient method [1-3], Newton method [4] linear programming [5,6] and Interior point method [7]. These traditional techniques rely on convexity to find the optimum solution. But due to the non-linear and non-convex nature of the OPF problem, the methods based on these assumptions do not guarantee to find the global optimum. Also, the discrete variables related to the tap changing transformer and shunt capacitors cannot be incorporated directly into the general optimal power flow programs. Recently Evolutionary Computation Techniques like Genetic Algorithm [8-9] and Evolutionary Programming [10] have been successfully applied to solve the OPF problems. After the Clean Air Act Amendments (Kyoto Protocol) in 1990, operating at minimum cost maintaining the security is no longer the sufficient criterion for dispatching electric power. Minimization of polluted gases such as NOx, SOx and COx in case of thermal generation power plants is also

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 167–175, 2011.
© Springer-Verlag Berlin Heidelberg 2011

becoming mandatory for the generation utilities in many countries. Hence, the minimization of pollution emission has to be considered along with the cost problem which makes OPF a multi-objective optimization problem [11-14]. Panigrahi et al [15] proposed a novel optimization approach for ELD problem using artificial immune system. The approach utilizes clonal selection principle and evolutionary approach wherein cloning of antibodies is performed followed by hypermutation.

In the proposed approach the OPF problem is formulated as non linear constrained multi objective problem where fuel cost and environmental impact are treated as competing objectives. In addition, GA operators were included in PSO algorithm to improve the effectiveness of the algorithm. In this paper, continuous variables are represented as floating point numbers and discrete variables are represented as integers. Further for effectiveness, crossover and mutation operators which can directly operate on floating point numbers and integers are used. The effectiveness of the proposed GA -PSO has been demonstrated through IEEE 30 bus system.

2 Problem Statement

List of Symbols

F_T	Total fuel cost;
P_{loss}	Network real power loss;
P_i, Q_i	Real and Reactive Powers injected into network at bus i;
G_{ij}, B_{ij}	Mutual conductance and susceptance between bus i and bus j;
G_{ii},B_{ii}	Self- conductance and susceptance of bus i;
P_{gi}, Qg	Real and Reactive power generation at bus i ;
Q_{Ci}	Reactive power generated by i^{th} capacitor bank;
t_k	Tap setting of transformer at branch k ;
V_i	Voltage magnitude at bus i;
V_j	Voltage magnitude at bus j;
θ_{ij}	Voltage angle difference between bus i and bus j;
S_l	Apparent power flow through the l^{th} branch;
g_k	Conductance of branch k;
N_B	Total number of buses;
N_{B-1}	Total Number of buses excluding slack bus;
N_{PQ}	Number of PQ buses;
N_g	Number of generator buses;
N_c	Number of capacitor banks;
N_T	Number of tap-Setting transformer branches;
N_l	Number of branches in the system.

A. Fuel Cost Minimization

In general, the optimal power flow problem is formulated as an optimization problem in which minimize a specific objective function is minimized while satisfying a number of equality and inequality constraints. Mathematically, this problem is stated as,

$$\text{Minimize } F_T = \sum_{i=1}^{N_g} F_i \tag{1}$$

Where $F_i = a_i P_{Gi}^2 + b_i P_{Gi} + c_i$ \$/hr

B. Emission Cost Minimization

The problem formulation is same as that of real power dispatch problem, but emission coefficients in place of fuel coefficients are used and dispatching is done by allocation of power generation across various generation units [16]. The problem formulations for various emissions are given below:

$$\text{Min } F(x) = \sum_{i=1}^{N_G} \alpha_i + \beta_i P_{Gi} + \gamma_i P_{Gi}^2 \tag{2}$$

Where α_i, β_i and γ_i are coefficients of generator emission characteristics and vary for NOx, SOx and COx Gases.

Subject to:
(i) Load flow constraints:

$$P_i - V_i \sum_{j=1}^{N_B} V_j (G_{ij} Cos\theta_{ij} + B_{ij} Sin\theta_{ij}) = 0, \quad i = 1,2,......N_{B-1} \tag{3}$$

$$Q_i - V_i \sum_{j=1}^{N_B} V_j (G_{ij} Sin\theta_{ij} - B_{ij} Cos\theta_{ij}) = 0, \quad i = 1,2,.....N_{PQ} \tag{4}$$

(ii) Voltage constraint:

$$V_i^{min} \le V_i \le V_i^{max} \qquad i \in N_B \tag{5}$$

(iii) Generator reactive power generation limit:

$$Qg_i^{min} \le Qg_i \le Qg_i^{max} \qquad i \in N_g \tag{6}$$

(iv) Reactive power generation limit of capacitor banks

$$Q_{Ci}^{\ min} < Q_{Ci} \le Q_{Ci}^{\ max} \qquad i \in N_C \tag{7}$$

(v) Transformer tap setting limit:

$$t_k^{min} \le t_k \le t_k^{max} \qquad k \in N_T \tag{8}$$

(vi) Transmission line flow limit

$$S_l \le S_l^{max} \qquad l \in N_l \tag{9}$$

The equality constraints given by Equations (3) and (9) are satisfied by running the power flow program. The active power generation (P_{gi}) (except the generator at the slack bus), generator terminal bus voltages (V_{gi}), transformer tap settings (t_k) and reactive power generation of capacitor bank (Q_{ci}) are the optimization variables and they are self-restricted by the optimization algorithm.

3 Proposed Hybrid GA Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a simple and efficient population based optimization method proposed by Kennedy and Eberhart [17]. Let x_i and v_i denote the positions and the corresponding flight speed (velocity) of the particle *i* in a continuous search space, respectively. The modified velocity and position of each particle can be calculated using the current velocity and the distance from the pbest and gbest as follows:

$$V_i^{k+1} = wV_i^k + c_1 \text{ rand}_1(.)(\text{pbest}_i - s_i^k) + c_2 \text{ rand}_2(.) .(\text{gbest} - s_i^k) \tag{10}$$

$$x_i^{(t+1)} = x_i^{(t)} + v_i^{(t+1)} \tag{11}$$

Where:

v_i^k: velocity of agent i at iteration k,
w: weighting function,
c_j : weighting factor,
rand : uniformly distributed random number between 0 and 1
s_i^k : current position of agent i at iteration k,
pbest_i : pbest of agent i,
gbest: gbest of the group.
v_i^{t+1} : velocity of agent i at iteration t+1
x_i^t : position of agent i at iteration t
x_i^{t+1} : position of agent i at iteration t+1

In the velocity updating process, the value of the parameters such as w, c1, c2 and k should be determined in advance. In the proposed GA-PSO for OPF problem, genetic operators were incorporated in the PSO algorithm which improves the effectiveness of the proposed algorithm. In the proposed algorithm, before population is updated, elitism was executed and best solutions from previous population were retained, while velocity and particle position were updated based on a component from GA algorithm, i.e. Mutation which is incorporated for updating particle position and generating new population. This was done because as PSO algorithm progresses, and convergence towards global best solution is attained, process of updating population is slowly halted, mutation according to fitness keeps on updating population in order to further optimize solution and converge towards even better solution.

4 PSO Implementation

While applying PSO for the OPF problem, the following issues need to be addressed:

- Problem Representation and
- Fitness evaluation.

4.1 Problem Representation

Implementation of PSO for a problem starts with the parameter encoding. Each individual in the population represents a candidate solution. With mixed form of representation, an individual in the PSO population for the OPF problem will look like the following:

$$\underbrace{97.5}_{P_{g2}}\ \underbrace{100.8}_{P_{g3}}\ \ldots\ \underbrace{250.70}_{P_{gn}}\quad \underbrace{0.981}_{V_{g1}}\quad \underbrace{0.970}_{V_{g2}}\ \ldots\ \underbrace{1.05}_{V_{gn}}\ \underbrace{3\,..}_{t_1}\ \underbrace{1\ \ldots}_{t_2}\ \underbrace{8}_{t_n}\quad \underbrace{2}_{Q_{c1}}\quad \underbrace{1\ \ldots..}_{Q_{c2}}\ \underbrace{8}_{Q_{cn}}$$

This coding scheme avoids the redundant value mapping or the introduction of artificial constraints which is necessary if binary coding is used to represent the transformer tap setting and the reactive power generation of capacitor. Also, with direct representation of the solution variables, the computer space required to store the population is reduced.

4.2 Evaluation Function

PSO searches for the optimal solution by maximizing a given fitness function, and therefore an evaluation function which provides a measure of the quality of the problem solution must be provided. In the OPF problem under consideration, the objective is to minimize the total fuel cost satisfying the constraints. With the inclusion of the penalty function the new objective function becomes,

$$Miinf = F_T + K_s(P_{sl} - P_{sl}^{\max})^2 + K_v \sum_{i=1}^{N_{PQ}} (V_i - V_i^{\lim})^2 + K_q \sum_{i=1}^{N_g} (Q_{gi} - Q_{gi}^{\lim})^2 + K_l \sum_{l=1}^{N_l} (S_l - S_l^{\lim})^2 \tag{12}$$

Where K_s, K_v, K_q, K_l and K_s are the penalty factors for the slack bus power output, bus voltage limit violation, generator reactive power limit violation, line flow violation and voltage stability limit violation respectively. The success of the approach lies in the proper choice of these penalty parameters. Since PSO maximizes the fitness function, the minimization objective function f is transformed to a fitness function to be maximized as,

Fitness = k/f where k is a large constant.

5 Simulation Results

The proposed GA based PSO approach has been applied to solve the economic emission optimal power flow problem in IEEE-30 bus system. The buses 10, 12, 15, 17, 20, 21, 23, 24 and 29 are identified for reactive power injection. The generator cost coefficients and the transmission line parameters are taken from [2]. The emission characteristics of Sox, NOx and COx are given in Appendix A1. The OPF algorithm solved using GA based PSO was implemented using the MATLAB program and was executed on Intel Core Duo 1.66GHz, with a 2 GB memory computer. Two different cases were considered for simulation, one minimizing fuel cost without considering the emission function and the next one to solve the OPF problem by including the emission function in the problem formulation. The results of these simulations are presented below.

Case 1: Minimizing Generation Cost (Single Objective Optimization)

In this case, PSO algorithm was applied to solve the optimal power flow with single objective function to minimize fuel cost by identifying the optimal control parameters using different PSO parameter settings. The minimum fuel cost obtained using PSO based OPF is 802.06 $/hr with no limit violations. In addition different combinations of GA operators were included in the OPF problem and are mentioned in Table 1. From this table, it is clear that crossover and mutation operators when incorporated into PSO algorithm provides better results than the other combinations.

Table 1. Different combinations of genetic operators in PSO algorithm

GA operators	P_m	Pc	Minimum fuel cost ($/hr)
Mutation	0.2	0	802.31
Mutation, elitism, selection & crossover	0.001	0.9	803.4
Elitism & mutation	0.001	-	802.02
Elitism, selection & mutation	0.0013	-	802.09
Elitism, crossover & mutation	0.05	1	802.35
Crossover &mutation	0.001	0.6	801.81

The optimal settings corresponding to the optimal combination of GA based PSO is given in Table 2. Further, the minimum cost solution obtained by the proposed approach is less than the values reported in literature for the IEEE 30-bus system and is shown in Table 3. This shows the powerfulness of the proposed algorithm to obtain the optimal solution in the OPF problem.

Table 2. Optimal control settings for GA-PSO for OPF problem

Control Variables	**Variable Setting**	**Control Variables**	**Variable setting**
P_1	176.37	T_{11}	3.0
P_2	48.88	T_{12}	0.0
P_5	21.47	T_{15}	2.0
P_8	21.98	T_{36}	0.0
P_{11}	12.03	Q_{c10}	5
P_{13}	12.00	Q_{c12}	5
V_1	1.0499	Q_{c15}	5
V_2	1.0366	Q_{c17}	5
V_5	1.0096	Q_{c20}	4
V_8	1.0171	Q_{c21}	5
V_{11}	1.0999	Q_{c23}	3
V_{13}	1.1000	Q_{c24}	5
		Q_{c29}	1
Cost	801.81 $/hr		

Table 3. Comparison of fuel cost

Algorithm	Fuel cost ($/hr)
Tabu Search [18]	802.50
Improved GA [14]	802.19
Modified DE [19]	802.37
PSO	**802.06**
Hybrid PSO	**801.81**

Table 4. Comparison of fuel cost and emission level with PSO and GA-PSO

Gas	**Nox**		**Sox**		**Cox**	
Optimization technique	PSO	GA based PSO	PSO	GA based PSO	PSO	GA based PSO
Fuel cost ($/hr)	849.511	848.467	855.546	847.710	872.0426	862.579
Emission (Kg/hr)	1263.7	1263.3	2892.3	2884.7	15043	15008

Case 2: Hybrid GA-PSO with minimizing Fuel Cost and Emission functions

In the proposed multi-objective problem, the problem was handled as a multi objective optimization problem with fuel cost and controlled emission as objectives to be minimized simultaneously. The fuel cost objective problem is combined with three combinations of polluted gases namely NOx, SOx and COx and the simulation results are given in Table 4. From this table, it is clear that multi objective GA-PSO provides better results than multi-objective PSO for all combinations of objective functions.

6 Conclusion

This paper has proposed an improved Particle Swarm Optimization incorporating genetic algorithm operators to solve the economic emission OPF problem. The performance of PSO with different combination of genetic operators has been explored. Simulation results on the IEEE 30-bus system are presented to illustrate the effectiveness of the proposed approach to solve the economic emission OPF problem. Hence by incorporating emission control into OPF problem, the proposed method converges fast, provides better outputs and also occupies less memory space. In future, MOPSO discussed in [20] will be considered to solve the multi-objective OPF problem.

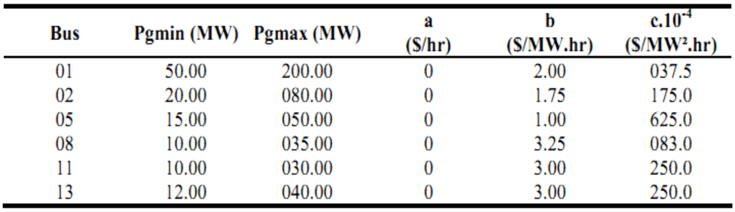
References

1. Dommel, H.W., Tinney, W.F.: Optimal power flow solutions. IEEE Transactions on Power Apparatus and Systems PAS-87 (10), 1866–1876 (1968)
2. Alsac, O., Scott, B.: Optimal load flow with steady state security. IEEE Transactions on Power Systems PAS 93(3), 745–751 (1974)
3. Lee, K.Y., Park, Y.M., Ortiz, J.L.: Optimal real and reactive power dispatch. Electric Power Systems Research 7, 201–212 (1984)
4. Sun, D.I., et al.: Optimal power flow by Newton approach. IEEE Transactions on PAS 103(10), 2864–2880 (1984)
5. Stott, B., Hobson, E.: Power system security control calculations using linear Programming. IEEE Transactions on Power Apparatus and Systems, PAS 97, 1713–1931 (1978)
6. Mangoli, M.K., Lee, K.Y.: Optimal real and reactive power control using linear Programming. Electric Power Systems Research 26, 1–10 (1993)
7. Momoh, J.A., Zhu, J.Z.: Improved interior point method for OPF problems. IEEE Transactions on Power Systems 14, 1114–1120 (1999)
8. Lai, L.L., Ma, J.T., Yokoyama, R., Zhao, M.: Improved genetic algorithm for optimal power flow under both normal and contingent operation states. Electrical Power and Energy Systems 9(5), 287–292 (1997)
9. Bauirtzis, A.G., et al.: Optimal power flow by Enhanced genetic algorithm. IEEE Transactions on Power Systems 14, 1114–1120 (1999)
10. Yuryevich, J., Wang, K.P.: Evolutionary Programming based optimal power flow algorithms. IEEE Transactions on Power Systems 14(4), 1245–1250
11. Deb, K.: Multi-objective evolutionary algorithm. John Wiley & Sons publications
12. Abido, M.A.: Multi-objective particle swarm optimization for environmental/economic dispatch problem. Electric Power System Research 79, 1105–1113 (2009)

13. Wang, L., Singh, C.: Stochastic combined heat and power dispatch based on multi-objective particle swarm optimization. International Journal of Electrical Power and Energy Systems 30, 226–234 (2008)
14. Devaraj, D., Preetha Roselyn, J.: Improved genetic algorithm for voltage security constrained optimal power flow problem. Int. Journal Energy Technology and Policy 5(4), 475–488
15. Panigrahi, B.K., Yadav, S.R., Agrawal, S., Tiwari, M.K.: Aclonal algorithm to solve economic load dispatch. Electric Power System Research 77, 1381–1389 (2007)
16. Bouktir, T., Labdani, R., Slimani, L.: Economic Power Dispatch of Power System With Pollution Control Using Multi objective Particle Swarm Optimization. Journal of Pure and Applied Sciences 4, 57–77 (2007)
17. Kennedy, J., Eberhart, R.: Swarm intelligence. Morgan Kaufmann publishers (2001)
18. Ongsakul, W., Tantimaporn, T.: Optimal power flow by improved evolutionary Programming. International Journal of Electric power Component and Systems (34), 79–95 (2006)
19. Sayah, S., Zehar, K.: Modified differential evolution algorithm for optimal power flow with non-smooth cost functions. Energy Conversion and Management 49, 3036–3042 (2008)
20. Zhou, A., Ru, B.-Y., Huiti, Zhao, S.-Z., Suganthan, P.N., Zhang, Q.: Multi-objective Evolutionary Algorithm:A survey of the state of the art. Swarm and Evolutionary Computation 1, 32–49 (2011)

Appendix

A1. Emission cost co-efficients



Bus	Pgmin (MW)	Pgmax (MW)	a ($/hr)	b ($/MW.hr)	$c.10^{-4}$ ($/MW^2.hr)
01	50.00	200.00	0	2.00	037.5
02	20.00	080.00	0	1.75	175.0
05	15.00	050.00	0	1.00	625.0
08	10.00	035.00	0	3.25	083.0
11	10.00	030.00	0	3.00	250.0
13	12.00	040.00	0	3.00	250.0

Application of Improved PSO Technique for Short Term Hydrothermal Generation Scheduling of Power System

S. Padmini[1], C. Christober Asir Rajan[2], and Pallavi Murthy[3]

[1] Department of Electrical and Electronics Engineering, SRM University, Chennai, India
[2] Department of Electrical and Electronics Engineering Pondichery University, Chennai, India
[3] Department of Electrical and Electronics Engineering, SRM University, Chennai, India
padminisp81@gmail.com, asir_70@hotmail.com,
pllv.murthy@gmail.com

Abstract. This paper addresses short-term scheduling of hydrothermal systems by using Particle Swarm Optimization (PSO) algorithm. Particle Swarm Optimization is applied to determine the optimal hourly schedule of power generation in a hydrothermal power system. The developed algorithm is illustrated for a test system consisting of one hydro and one thermal plant respectively. The effectiveness and stochastic nature of proposed algorithm has been tested with standard test case and the results have been compared with earlier works. It is found that convergence characteristic is excellent and the results obtained by the proposed method are superior in terms of fuel cost .

Keywords: Hydrothermal Scheduling, Particle Swarm Optimization.

1 Introduction

Now a days, because of the increasing competition in power market, scheduling the Hydro and Thermal energy in the most economic manner has become an important task in modern power systems. The main objective of hydrothermal operation is to minimize the total system operating cost, represented by the fuel cost for the systems thermal generation subject to the operating constraints of hydro and thermal plant over the optimization interval. In short range problem the water inflows is considered fully known and is constrained by the amount of water available for draw down in the interval. The short term hydro thermal scheduling problems have been solved by various methods. These methods which have been reported in the literature includes classical methods such as Langrage Multiplier Gradient Search and Dynamic programming stochastic search algorithm such as simulated annealing (SA) [6], Genetic algorithm (GA) [3], Evolutionary Programming (EP) [7,8] and Particle Swarm Optimization (PSO) [2,9,10,11]. The PSO technique can generate high-quality solution within shorter calculation time and more stable convergence characteristic than other stochastic methods. Many researches are still for proving its potential in solving complex power system problems.

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 176–182, 2011.
© Springer-Verlag Berlin Heidelberg 2011

2 Problem Formulation

2.1 Objective Function

The main objective of Hydro Thermal Scheduling is minimizing the thermal generation cost by satisfying the hydro and thermal constraints. Hydro thermal scheduling is the optimization of a problem with non-linear objective function, the objective function to be minimized can be written as:

The problem formulation is same as that of real power dispatch problem, but emission coefficients in place of fuel coefficients are used and dispatching is done by allocation of power generation across various generation units [16]. The problem formulations for various emissions are given below:

$$\min(f)=\sum_{j=1}^{N} n_j f_j\left(PT_j\right) \tag{1}$$

Where

j = 1, 2, 3 ....N schedule interval

$f_j\left(PT_j\right)$ is the thermal generation cost in \$/h

n_j is the number of hours in j^{th} interval

Subject to:

Power balance constraint

$$\left(P_{dj}+P_{Loss_j}\right)-\left(PH_j+PT_j\right)=0 \tag{2}$$

The transmission loss is given by

$$P_{Loss}=k\left(PH_j\right) \tag{3}$$

The hydro generation is consider to be a function of discharge rate only

$$q_j=g\left(PH_j\right) \tag{4}$$

Discharge rate limits

$$q_{\max}>q_j>q_{\min} \tag{5}$$

Thermal generation limits

$$PT_{\max}>PT_j>PT_{\min} \tag{6}$$

Hydro generation limits

$$PH_{\max}>PH_j>PH_{\min} \tag{7}$$

Reservoir volume

$$Volume_{j+1}=Volume_j+n_j\left(r_j-q_j-s_j\right) \tag{8}$$

Where

$n\,j$ is the number of hours in jth interval

rj is the water inflow rate in jth interval

$q\,j$ is the water discharge rate in jth interval

$s\,j$ is the water spillage rate in jth interval

Reservoir storage limits

$$Volume_{\max} > Volume_j > Volume_{\min} \tag{9}$$

3 Particle Swarm Optimization

Particle swarm optimization (PSO), first introduced by Kennedy and Eberhart in 1995, is one of the modern heuristic optimization algorithms. Among all the stochastic search algorithms PSO gives the reasonable solution in a short CPU time. The application of PSO technique into nonlinear discontinuous constrained power system problems such as Economic Dispatch (ED), Unit Commitment (UC), state Estimation (SE), Load Forecasting (LF), Optimal Power Flow (OPF), etc have been reported by several authors. This method has been developed under the scope of artificial life where PSO is inspired by the natural phenomenon of fish schooling or bird flocking. PSO is initialized with a group of random particles and then searches for optima by updating generation. In every iteration each particle is updated by two best values. First one is based on its own best exploration called pbest. Another best value is based on best swarm overall experience called *gbest*. After finding two best values the particle velocity and positions are updated by following equations.

$$V_j^{k+1} = WV_{j^k}^{k} + c_1 rand_1()\left(pbest - x_j^k\right) + c_2 rand_2()\left(gbest - x_j^k\right) \tag{10}$$

$$x_j^{k+1} = x_j^k + K * V_{ji}^{k+1} \tag{11}$$

Where

$V_j^k =$ Velocity of individual particle (i) at iteration k

W = Weight parameter

$$W = W_{\max} - \frac{W_{\max} - W_{\min}}{iter_{\max}} * iter \tag{12}$$

c_1 & c_2 : Acceleration Constant

$c_1 = 2.5 = c_2$

$rand_1$( ) & $rand_2$( ) : Uniform random number in range [0,1]

p_{best} : Previous best value

g_{best} : Global best value

K : Constriction factor

4 Proposed PSO Algorithm for Hydrothermal Scheduling

The various possible particles are Hydro generation, thermal generation, water discharge rate and reservoir volume for hydro thermal scheduling problem. One can select any of the above as particle. The work shows that selecting discharge as a particle gives the best result. The proposed PSO based HTS algorithm with reservoir volume as particle is presented here.

Step 1: Input parameters of the system and specify the upper and lower boundaries of each variable.

Step 2: Initialize randomly the particles of the population.

Step 3: Let qj be the dishcarge rate denoting the particles of population to be evolved.It is the discharges of turbines of reservoirs at various intervals.Then knowing the hydro discharges, storage volumes of reservoirs V_j are calculated using Eq.(8).Then P_H *and* P_T is calculated for all the intervals.

Step 4: Compare each particle with its Pbest.The best evaluation value among the Pbest is denoted as gbest.

Step 5: Update the iteration count and update the velocity and particle position uisng Eq. (10) and Eq. (11)respectively.

Step 6: Each particle is evaluated according to its updated position, only when satisfied by all constraints. If the evaluation value of each particle is better than the previous *Pbest*. The current value is set to be *Pbest*. If the best *Pbest* is better than *gbest*, the value is set to be *gbest*.

Step 7: If the stopping criterion is reached, then print the result and stop; otherwise repeat steps 2–6.

5 Test System and Results

In this study, a test system consisting of one hydro unit and one thermal unit respectively. The entire optimization period is three days and it has been divided into 6 intervals and each interval is of 12 hours. Three days' twelve hour based load curve is given in Table 1.

Table 1. Load Demand

Interval	**Day**	**Time Interval**	**Demand (MW)**
1	1	0-12h	1200
2	1	12-24h	1500
3	2	0-12h	1100
4	2	12-24h	1800
5	3	0-12h	950
6	3	12-24h	1300

Water discharge rate

$$q_j = 330 + 4.97 * PHj \quad \text{if} \quad 0 < PHj < 1000 \quad acre-ft/h \tag{13}$$

$$q_j = 5300 + 12\,(PHj - 1000) + 0.05\,(PHj - 1000)^2 \tag{14}$$

$$\text{if} \quad PHj > 1000 \quad acre-ft/h$$

$$q_j = 0 \quad \text{if} \quad PHj = 0$$

$$\text{Volume}_{start} = 100000 \; acre-ft$$

$$\text{Volume}_{End} = 60000 \; acre-ft$$

$$\text{Volume}_{mix} = 120000 \; acre-ft$$

$$\text{Volume}_{min} = 60000 \; acre-ft$$

The objective function, hydro constraints and thermal constraints are given below.

The cost characteristic of thermal plant is given by:

$$f\,(PT_j) = 0.000184\, PH_j^2 + 9.2\, PH_j + 575 \;\; \$/h \tag{15}$$

$$150 < PT_j < 1500 \quad MW$$
$$0 < PT_j < 1100 \quad MW$$

The reservoir is considered to have a constant water inflow (r_j) of 2000 acre-ft/h. The spillage rate $j\ s$ is considered to be zero. Transmission loss is taken to be negligibly small.

In applying the proposed PSO algorithm for the test system, the appropriate values of population size NP and maximum iteration number Nmax are set to the values of 30 and 100 , respectively.Table-2 gives the results of PSO by considering water discharge as a particle.

Table 2. Results of Proposed Method

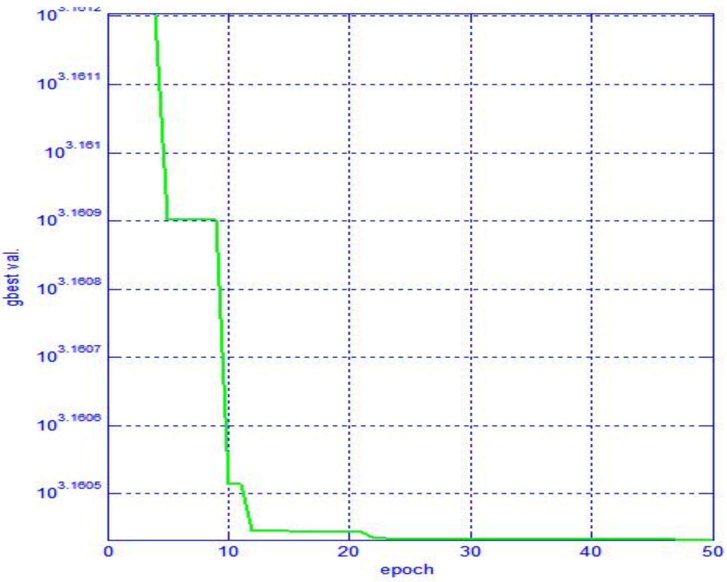
Interval	**PT (MW)**	**PH (MW)**	**Volume (acre ft)**	**Discharge Rate (acreft/hr)**	**Cost ($)**
1	812.54	387.45	96931.91	2255.674	606423.16
2	801.58	698.41	75318.31	3801.133	
3	1100	0	99314.31	0	
4	804.72	995.27	59996.04	5276.52	
5	950	0	83996.04	0	
6	561.56	738.43	59996.04	4000	

Table 3. Comparison of Results

S.No	Author	Year	Method	Cost
1	Wood.A.J	1984	GS	709877.38
2	Sinha.et.al	2003	IFEP	709862.05
3	Wong.K.P.et.al	1994	SA	709874.36
4	Sinha.et.al	2006	GAF	709863.70
5	Sinha.et.al	2006	CEP	709862.65
6	Sinha.et.al	2006	FEP	709864.59
7	Sinha.et.al	2006	PSO	709862.04
8	D.S.Suman.et.al	2006	Hybrid EP	703180.26
9	Proposed	2011	PSO	606423.16

The result of the proposed algorithm has been compared with the results of earlier works in Table-3. The proposed algorithm was more successful in reducing the cost searching the entire search space than all other earlier methods. All the earlier work results have taken from references [1, 3, 6, 7, 8 and 9]. Table-3 gives the comparison of earlier works and proposed method.

The convergence characteristics of the proposed method is given in figure-1.



6 Conclusion

In this paper, a particle swarm optimization (PSO) based algortihm has been proposed for solving Short term hydrothermal scheduling problem by taking the discharge rate as particle.The results shows that best optimal solutions can be obtained by particle swarm optimization method,when compared with already existing techniques.A variant of particle swarm optimizer can be used whereby all other particles'historical best information can be used to update the particle's velocity.This strategy enables the diversity of the swarm to be preserved to discourage premature convergence.This can be used as the future work to yield better convergence .

References

1. Wood, A.J., Wollenberg, B.F.: Power Generation, Operation and Control. Operation and Control. John Wiley and Sons, New York (1984)
2. Umayal, S.P., Kamaraj, N.: Stochastic Multi Objective Short term Hydrothermal Scheduling Using Particle Swarm Optimization (2005)
3. Ferrero, R.W., Rivera, J.F., Shahidehpour, S.M.: A dynamicprogramming two-stage algorithm for long-termhydrothermal scheduling of multireservoir systems. IEEE Transactions on Power Systems 13(4), 1534–1540 (1998)
4. Chang, W.: Optimal Scheduling of Hydrothermal System Based on Improved Particle Swarm Optimization
5. Samudi, C., Das, G.P., Ojha, P.C., Sreeni, T.S., Cherian, S.: Hydro thernal Scheduling using Particle Swarm Optimization (2008)
6. Wong, K.P., Wong, Y.W.: Short-term hydrothermal scheduling, part-I: Simulated. annealing approach. IEE Proc., Part- C 141(5), 497–501 (1994)
7. Sinha, N., Chakrabarti, R.: Fast Evolutionary Programming Techniques For Short-Term Hydrothermal Scheduling. IEEE Trans. PWRS 18(1), 214–219 (2003)
8. Suman, D.S., Nallasivan, C., Henry, J., Ravichandran, S.: A Novel Approach for Short-Term Hydrothermal Scheduling Using Hybrid Technique. In: IEEE Power India Conference, April 10-12 (2006)
9. Sinha, N., Lai, L.-L.: Meta Heuristic Search Algorithms for Short-Term Hydrothermal Scheduling. In: International Conference on Machine Learning and Cybernetics, Dalian (2006)
10. Hotaa, P.K., Barisala, A.K., Chakrabarti, R.: An improved PSO technique for short-term optimal hydrothermal scheduling. Electric Power Systems Research 79(7), 1047–1053 (2009)
11. Liang, J.J., Qin, A.K., Suganthan, P.N., Baskar, S.: Comprehensive Learning Particle Swarm Optimizer for Global Optimization of Multimodal Functions. IEEE T. on Evolutionary Computation 10(3), 281–295 (2006)

Multi-objective Workflow Grid Scheduling Based on Discrete Particle Swarm Optimization

Ritu Garg and Awadhesh Kumar Singh

Computer Engineering Department, National Institute of Technology,
Kurukshetra, Haryana, India
ritu.59@gmail.com, aksinreck@rediffmail.com

Abstract. Grid computing infrastructure emerged as a next generation of high performance computing by providing availability of vast heterogenous resources. In the dynamic envirnment of grid, a schedling decision is still challenging. In this paper, we present efficient scheduling scheme for workflow grid based on discrete particle swarm optimization. We attempt to create an optimized schedule by considering two conflicting objectives, namely the execution time (makespan) and total cost, for workflow execution. Multiple solutions have been produced using non dominated sort particle swarm optimization (NSPSO) [13]. Moreover, the selection of a solution out of multiple solutions has been left to the user. The effectiveness of the used algorithm is demostrated by comparing it with well known genetic algorithm NSGA-II. Simulation analysis manifests that NSPSO is able to find set of optimal solutions with better convergence and uniform diversity in small computation overhead.

1 Introduction

With the rapid development of networking technology, grid computing [4] has emerged as a promising distributed computing paradigm that enables large-scale resource sharing and collaboration. One of the key challenges of heterogeneous systems is the scheduling problem. Scheduling of computational tasks on the Grid is a complex optimization problem, which may require consideration of several scheduling criteria. Usually, the most important criterion is the task execution time, cost of running a task on a machine, reliability, resource utilization etc.

The optimization of scheduling problem is NP-complete, so numerous heuristic algorithms have been proposed in literature [3]. Many heuristics have also been proposed for workflow scheduling in order to optimize a single objective [9], [10], [16], [20]. To achieve better solution quality, many meta-heuristic methods have been presented for job scheduling such as simulated annealing (SA) [2], genetic algorithm (GA) [15], ant colony optimization (ACO) [14] and tabu search[6].

Defining the multiple objectives for the task scheduling problem for generating efficient schedules at reduced computational times are of research interest in the recent days. For multi objective optimization to independent task scheduling [1] used the fuzzy particle swarm optimization and [11] used the discrete particle swarm optimization. These methods combine the multiple objectives into scalar cost function

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 183–190, 2011.
© Springer-Verlag Berlin Heidelberg 2011

using the weight factors, which convert the problem into single objective problem prior to optimization. Generally, it is very difficult to accurately select these weights as small perturbations in weights leads to different solutions.

Hence, in this paper we introduced the multi objective optimization approach based on discrete particle swarm optimization to generate Pareto optimal solutions, which is the set consisting of all non-dominated solutions. A solution is called non-dominated solution if it is best at least in one objective with respect to others. Pareto optimal solutions are preferred to single solution in real life applications. In this paper, we considers the two objectives for task scheduling keeping in view the tradeoff between two conflicting objectives of minimizing the makespan and total cost under the specified deadline and budget constraint. Towards the goal of obtaining Pareto optimal set, we applied non dominated sort PSO (NSPSO) [13] as it performs better against a set of well known test functions that are presumed difficult. Rest of the paper is organized as follows. Section 2 specifies the problem definition. In section 3, describes the formulation of discrete particle swarm optimization for multi objective workflow grid scheduling. In section 4, we explained the procedure of non dominated sort particle swarm optimization algorithm which is used to obtain Pareto optimal solutions. Section 5 discusses the simulation analysis and finally section 6 gives the conclusion.

2 Problem Definition: Workflow Grid Scheduling

We define workflow Grid scheduling as the problem of assigning different precedence constraint tasks in the workflow to different available grid resources. We model the application as a task graph: Let G = (V, E) be a directed acyclic graph (DAG), with V as the set of n tasks $t_i \in V$, $1 \le i \le n$ and E is the set of edges representing precedence constraint among the tasks $e_{ij} = (t_i, t_j) \in E$, $1 \le i \le n$, $1 \le j \le n$, $i \ne j$. Associated to each edge is the amount of data required to send from task t_i to t_j if they are not executed on the same resource. Let set R represent the m number of resources which are available in the grid and resource $r_j \in R$ is associated with two values: Time and cost of executing the task on resource r_j. Every task t_i has to be processed on resource r_j until completion. In our work, scheduling solution is represented as the task assignment string corresponding to the scheduling order string. Task assignment string is the allocation of each task to the available time slot of the resource capable of executing the task, and the scheduling order string encodes the order to schedule tasks. The ordering of tasks in the scheduling order string must satisfy the task dependencies. The execution optimization problem is to generate task assignment string S, which maps every t_i onto a suitable r_j to achieve the multi objective below:

$$\text{Minimize Time}(S) = \max \mathit{time}(t_i) \text{ where } t_i \in V \text{ and } 1 \le i \le n \quad (1)$$

$$\text{Minimize Cost}(S) = \sum \mathit{cost}(t_i) \quad \text{where } t_i \in V \text{ and } 1 \le i \le n \quad (2)$$

$$\text{Subject to} \quad \text{Cost}(S) < B \text{ and Time}(S) < D$$

Where B is the cost constraint (Budget) and D is the time constraint (Deadline) required by users for workflow execution.

3 Discrete Particle Swarm Optimization for Workflow Grid Scheduling

In this paper, we used the version of discrete particle swarm optimization (DPSO) [11] to solve the problem of workflow grid scheduling. PSO is a self adaptive global search optimization technique introduced by Kennedy and Eberhart [12] and it relies on the social behavior of the particles. In every generation, each particle adjusts its trajectory based on its best position (local best) and the position of the best particle (Global best) of the entire population. One of the key issues in designing successful PSO algorithm is the representation step, i.e. finding a suitable mapping between problem solution and PSO particle. For optimization of workflow grid scheduling problem, solution is represented as task assignment string (S) as mentioned in section 2. To represent S, we setup an n dimension search space corresponding to n number of tasks and each dimension represents the discrete value corresponding to m number of resources.

Here, solutions or task assignment strings are encoded as m×n matrix, called position matrix where m is the number of available resources and n is the number of tasks. Let X_k is the position matrix of k^{th} particle then

$$X_k(i,j) \in \{0,1\}\ (\forall\ i,j),\ i \in \{1,2,...m\},\ j \in \{1,2,...,n\}. \tag{3}$$

where X_k (i, j) = 1 means that j^{th} task is performed by i^{th} resource. Hence, in each column of the matrix only single element is 1 and others are 0. For example, Fig.1. shows the mapping between one possible task assignment strings to the particle position matrix in PSO domain.

Task Assignment String

{[T1 : R3], [T2 : R1], [T3 : R3], [T4 : R2], [T5 : R1], [T6 : R3]}

Particle Position Matrix

	T1	T2	T3	T4	T5	T6
R1	0	1	0	0	1	0
R2	0	0	0	1	0	0
R3	1	0	1	0	0	1

Fig. 1. Mapping of Task assignment string to Particle Position matrix

Velocity of each particle is again an m×n matrix whose elements are in range $[-V_{max}, V_{max}]$. If V_k is the velocity matrix of k^{th} particle, then:

$$V_k(i,j) \in [-V_{max}, V_{max}],\ (\forall\ i, j),\ i \in \{1,2,...m\},\ j \in \{1,2,...,n\}. \tag{4}$$

Also, Pbest and Gbest are m×n matrices and their elements assume value 0 or 1 as in the case of position matrices. $Pbest_k$ represents the best position that k^{th} particle has visited since the initial time step and $Gbest_k$ represents the best position that k^{th} particle and its neighbors have visited since the algorithm was initiated. For updating $Pbest_k$ and $Gbest_k$ in each time stamp we are using the non dominated sort multi objective PSO algorithm as mentioned by procedure of NSPSO in section 4.

For particle updating, we are first updating velocity matrix according to (5) and then finally position matrix is updated using (6).

$$V_k^{(t+1)}(i,j) = \omega . V_k^{(t)}(i,j) + c_1 r_1 (Pbest_k^{(t)}(i,j) - X_k^{(t)}(i,j)) + c_2 r_2 (Gbest_k^{(t)}(i,j) - X_k^{(t)}(i,j)) \tag{5}$$

$$X_k^{(t+1)}(i,j) = \begin{cases} 1, & if\ V_k^{(t+1)}(i,j) = \max\left\{V_k^{(t+1)}(i,j)\right\} \forall\ i \in \{1,2,\ldots m\} \\ 0, & otherwise \end{cases} \tag{6}$$

Using equation (6), each column of position matrix, the value 1 is assigned to the element whose corresponding element in velocity matrix has maximum value in its corresponding column. If in a column of velocity matrix there are more than one element with max value, then one of these elements is selected randomly and 1 is assigned to its corresponding element in the position matrix.

A particle represented as position matrix X_k is formulated from task assignment string (S). Initially, S representing resource on which a task will execute is defined randomly. The fitness functions Ftime(S) and Fcost(S) are formed in order to evaluate individuals according to makespan and cost of the schedule respectively. These fitness functions are calculated from Equation (1) and (2) by adding the penalty. On the violation of deadline and budget constraints, penalty is added respective to objective function, otherwise not.

4 Multi Objective Optimization Algorithm Used

To optimize workflow grid scheduling under two conflicting objective of makespan and total cost, we are using non dominated sort particle swarm optimization (NSPSO) [13] approach. NSPSO extends the basic form of PSO by making a better use of particle's personal bests and offspring for effective non-domination comparisons. The steps of basic NSPSO procedure are as follows:

```
NSPSO Procedure
1. Create and initialize m×n dimensional swarm with N
   particles randomly. The initial velocity for each
   particle is also initialized randomly but in the
   range [$-V_{max}$, $V_{max}$]. The personal best position of each
   particle ($Pbest_k$), is set to $X_k$.
2. Evaluate each particle in the swarm.
3. Apply non-dominated sorting on the particles.
4. Calculate crowding distance of each particle.
5. Sort the solutions based on decreasing order of
   crowding distance.
```

```
6. Select randomly $Gbest^{(t)}$ for each particle from a
   specified top part (e.g. top 5%) of the first front
   F1;
7. Calculate the new velocity $V^{(t+1)}$ for all particles
   based on Equation (5) and new position $X^{(t+1)}$ from
   Equation (6) using the determined $Gbest^{(t)}$ and $Pbest^{(t)}$.
8. Create a new population of size 2N by combining the
   new position and their personal best, $X^{(t+1)} \cup Pbest^{(t)}$.
9. Apply non-dominated sorting on 2N particles and
   calculate the crowding distance for each particle.
10. Generate a new set of N solutions by selecting
    solutions from non-dominated fronts F1, F2 and so on
    using the crowding distance. The N solutions form the
    personal best for the next iteration.
11. Go to step 2 till the termination criteria is met.
```

5 Simulation Results and Discussion

We used GridSim [5] toolkit in our experiment to simulate the scheduling of workflow tasks. GridSim is a java based toolkit for modeling and simulation of resource and application scheduling in large-scale parallel and distributed computing environment such as Grid.

In our test environment, we simulated the balanced workflow consisting of 20 tasks on 8 virtual resources and these resources are maintained by different organizations in the grid. Links between resources are established through a router so that direct communication can take place between resources. Computational rating (Million instructions per second) and computational cost (in dollars) of each resource is generated with non-uniform distribution. Number of data units required by one task from another task in the workflow is also generated non-uniformly.

In order to generate valid schedule which can meet both deadline and budget constraints specified by the user, two algorithms HEFT [10] and Greedy Cost were used to make deadline and budget effectively. HEFT is a time optimization scheduling algorithm in which workflow tasks are scheduled on minimum execution time heterogeneous resources irrespective of utility cost of resources. So HEFT gives minimum makespan ($Time_{min}$) and maximum total cost ($Cost_{max}$) of the workflow schedule. Greedy Cost is a cost optimization scheduling algorithm in which workflow tasks are scheduled on cheapest heterogeneous resources irrespective of the task execution time. Thus Greedy Cost gives maximum makespan ($Time_{max}$) and minimum total cost ($Cost_{min}$) of the workflow schedule Thus Deadline (D) and Budget (B) are specified as:

$$D = Time_{max} – 0.1(Time_{max} - Time_{min}) \tag{7}$$

$$B = Cost_{max} – 0.1(Cost_{max} - Cost_{min}) \tag{8}$$

To measure the effectiveness and validity of NSPSO algorithm for workflow grid scheduling problem, we have implemented a probabilistic GA based technique known as non dominated sort genetic algorithm (NSGA-II) [7]. To implement the NSGA-II we have taken binary tournament selection, two point crossover and replacing mutation.

Each experiment was repeated 20 times with different random population initialization. Initial population was seeded with two solutions obtained by heuristics namely LOSS-II and modified GAIN-II. LOSS-II and modified GAIN-II were used to generate a solution with minimum makespan (total time) while meeting budget constraint and minimum total cost while meeting deadline constraint respectively. This is done because population with seeding which contains two already optimized boundary solutions gives good and fast convergence with better spread rather than generating all solutions randomly [20]. In order to compare the performance of algorithms, we have run the algorithms over 200 generations with the initial population size of 50. The Pareto optimal solutions obtained with NSPSO and NSGA-II for bi-objective workflow grid scheduling problem are shown in Fig. 2. From a typical run shown in Fig. 2 we can see that most of the solutions obtained with NSPSO are lie on the better front as compared to NSGA-II while preserving uniform diversity between solutions.

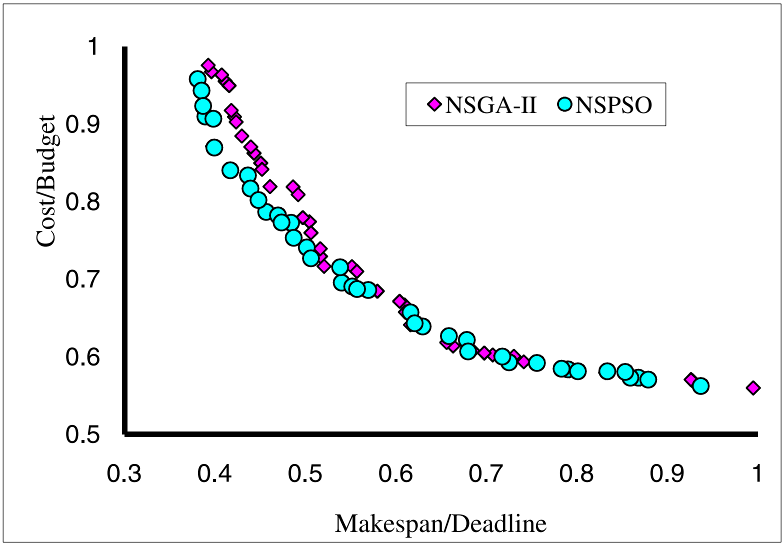


Fig. 2. Obtained Pareto Optimal Solutions with NSPSO and NSGA-II

5.1 Performance Evaluation: GD, Spacing

For the performance comparison between NSPSO and NSGA-II we conducted our experiment over 20 runs and then average of these runs has been taken for evaluation. To measure the quality of evolutionary algorithms, we used two metrics Generational Distance (GD) and Spacing [8]. GD is the well known convergence metric to evaluate the quality of an algorithm against the reference front P*. The reference front P* was obtained by merging solutions of algorithms considered. On the other side, Spacing metric was used to evaluate diversity among solutions. The results obtained with these metrics for each algorithm are depicted in Table 1. The average value of the GD and spacing metric corresponding to NSPSO is less as compared to other algorithm considered i.e, NSGA-II. The result confirms the better convergence towards the real Pareto optimal front. Further, the low value of standard deviation shows that algorithm converges almost in every execution.

Table 1. GD, Spacing Metric Results for the algorithms used

		NSGA-II	NSPSO
GD Metric	Avg	0.0231861	**0.021173**
	Std. Dev.	0.001567	0.001480
Spacing Metric	Avg	0.042911	**0.040233**
	Std. Dev.	0.003558	0.003324

6 Conclusion and Future Work

The current work emphasizes on the planning and optimizing the workflow scheduling in the grid. In this paper, we have used a version of discrete particle swarm optimization to represent the scheduling of workflow grid. The multi-objective non dominated sort particle swarm optimization (NSPSO) approach has been used to find solutions in the entire Pareto optimal front in order to minimize the two conflicting objectives of makespan and total cost. The performance of NSPSO algorithm has been evaluated in comparison with NSGA-II algorithm. The simulation results exhibit that NSPSO is a better compromised multi-objective optimization algorithm for workflow grid task scheduling in terms of convergence towards true Pareto optimal front and uniformly distributed solutions with small computation overhead. In future we plan to evaluate the scheduling approach using more than two objectives simultaneously. We will also apply more advanced algorithms like MOPSO [18], 2LB-MOPSO [17], fuzzy dominance based MOPSO [19] etc to obtain the optimal solutions for workflow grid scheduling problem.

References

1. Abraham, A., Liu, H., Zhang, W., Chang, T.G.: Scheduling Jobs on Computational Grids Using Fuzzy Particle Swarm Algorithm, pp. 500–507. Springer, Heidelberg (2006)
2. Attiya, G., Hamam, Y.: Task allocation for maximizing reliability of distributed systems: A simulated annealing approach. Journal of Parallel and Distributed Computing 66, 1259–1266 (2006)
3. Braun, T.D., Siegal, H.J., Beck, N.: A comparision of Eleven Static Heuristics for Mapping a Class of Independent Tasks onto Heterogeneous Distributed Computing Systems. Journal of Parallel and Distributed Computing 61, 810–837 (2001)
4. Buyya, R., Venugopal, S.: A Gentle Introduction to Grid Computing and Technologies. CSI Communications 29, 9–19 (2005)

5. Buyya, R., Murshed, M.: GridSim: A Toolkit for Modeling and Simulation of Grid Resource Management and Scheduling, vol. 14, pp. 1175–1220 (2002), `http://www.buyya.com/gridsim`
6. Chen, W.H., Lin, C.S.: A hybrid heuristic to solve a task allocation problem. Computers & Operations Research 27, 287–303 (2000)
7. Deb, K., Pratap, A., Aggarwal, S., Meyarivan, T.: A Fast Elitist Multi-Objective Genetic Algorithm. In: Deb, K., Rudolph, G., Lutton, E., Merelo, J.J., Schoenauer, M., Schwefel, H.-P., Yao, X. (eds.) PPSN 2000. LNCS, vol. 1917, pp. 849–858. Springer, Heidelberg (2000)
8. Deb, K., Jain, S.: Running Performance Metrics for Evolutionary Multi-objective Optimization. In: Proceedings of Simulated Evolution and Learning (SEAL 2002), pp. 13–20 (2002)
9. Wieczorek, M., Prodan, R., Fahringer, T.: Scheduling of Scientific Workflows in the ASKALON Grid Environment. SIGMOD 34(3), 56–62 (2005)
10. Haluk, T., Hariri, S., Wu, M.Y.: Performance-Effective and Low-Complexity Task Scheduling for Heterogeneous Computing. IEEE Transactions on Parallel and Distributed Systems 13, 260–274 (2002)
11. Izakian, H., Tork Ladani, B., Zamanifar, K., Abraham, A.: A Novel Particle Swarm Optimization Approach for Grid Job Scheduling. In: Prasad, S.K., Routray, S., Khurana, R., Sahni, S. (eds.) ICISTM 2009. CCIS, vol. 31, pp. 100–109. Springer, Heidelberg (2009)
12. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: Proceedings of IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948 (1995)
13. Li, X.: A Non-dominated Sorting Particle Swarm Optimizer for Multi- objective Optimization. In: Proceeding of Genetic and Evolutionary Computation Conference 2003 (GECCO 20003), Chicago, USA (2003)
14. Ritchie, G., Levine, J.: A fast, effective local search for scheduling independent jobs in heterogeneous computing environments, Technical report, Centre for Intelligent Systems and their Applications, School of Informatics, University of Edinburgh (2003)
15. Subrata, R., Zomaya, Y.A., Landfeldt, B.: Artificial life techniques for load balancing in computational grids. Journal of Computer and System Sciences 73, 1176–1190 (2007)
16. Tsiakkouri, E., Sakellariou, R., Zhao, H., Dikaiakos, M.D.: Scheduling Workflows with Budget Constraints. In: CoreGRID Integration Workshop Pisa, Italy (2005)
17. Zhao, S.Z., Zhao, P.N.: Two-lbests Based Multi-objective Particle Swarm Optimizer. Engineering Optimization 43, 1–17 (2011)
18. Coello, C.A.C., Pulido, G., Lechuga, M.: Handling multi-objective with particle swarm optimization. IEEE Trans. Evol. Comput. 8(3), 256–279 (2004)
19. Praveen, K., Das, S., Welch, S.M.: Multi-Objective Hybrid PSO Using ε-Fuzzy Dominance. In: Proceeding of Genetic and Evolutionary Computation Conference (GECCO 2007), London, UK (2007)
20. Yu, J., Buyya, R.: Scheduling Scientific Workflow Applications with Deadline and Budget constraints using Genetic Algorithms. Scientific Programming Journal 14(1), 217–230 (2006)

Solution of Economic Load Dispatch Problem Using Lbest-Particle Swarm Optimization with Dynamically Varying Sub-swarms

Hamim Zafar[1], Arkabandhu Chowdhury[1], and Bijaya Ketan Panigrahi[2]

[1] Dept. of Electronics and Telecommunication Engg., Jadavpur University, Kolkata, India
[2] Senior Member, *IEEE,* Dept. of Electrical Engineering, IIT Delhi, New Delhi, India

Abstract. This article presents an efficient optimization approach to solve constrained Economic Load Dispatch (ELD) problem using a 'Lbest-Particle Swarm Optimization with Dynamically Varying Sub-swarms' (LPSO-DVS). The proposed method is found to give optimal results while working with constraints in the ELD, arising due to practical limitations like dynamic operation constraints (ramp rate limits) and prohibited zones and also accounts valve point loadings. Simulations performed over various systems with different number of generating units with the proposed method have been compared with other existing relevant approaches. Experimental results support the claim of proficiency of the method over other existing techniques in terms of robustness, fast convergence and, most importantly its optimal search behavior.

1 Introduction

Economic load dispatch (ELD) is one of the most important problems involving optimization process in case of electric power system operation. This constrained optimization problem in power systems has the objective of minimizing fuel cost of generating units so as to accomplish optimal generation dispatch among operating units and in return satisfying the system load demand, generator operation constraints with ramp rate limits and prohibited operating zones. Over the years several endeavors have been made to solve this problem, the conventional methods include Lambda iteration method [1][2], base point and participation factors method [1][2]. This non-convex problem is very complex and cannot be solved by the traditional methods, with good result. Recently, different heuristic approaches have been proved to be effective with promising performance. These include evolutionary programming (EP) [3], simulated annealing (SA) [4], genetic algorithm (GA) [5], differential evolution (DE) [6], particle swarm optimization (PSO) [7], etc. The PSO algorithm has been empirically shown to perform well on many optimization problems. However, it may easily get trapped in a local optimum when solving complex multimodal problems. In order to improve PSO's performance on ELD problems, we present the Lbest-Particle Swarm Optimization with Dynamically Varying Sub-swarms (LPSO-DVS) utilizing a new learning strategy.

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 191–198, 2011.
© Springer-Verlag Berlin Heidelberg 2011

2 Problem Description

Economic Load Dispatch (ELD) problem is one of the different non-linear programming sub-problems of unit commitment. Two alternative models for ELD are detailed below.

2.1 ELD Formulation with Smooth Cost Function

The objective function corresponding to the production cost can be approximated to be a quadratic function of the active power outputs from the generating units. Symbolically, it is represented as

Minimize
$$F_t^{cost} = \sum_{i=1}^{N_G} f_i(P_i) \tag{1}$$

where
$$f_i(P_i) = a_i P_i^2 + b_i P_i + c_i, \quad i = 1,2,3,\ldots,N_G \tag{2}$$

is the expression for cost function corresponding to i^{th}generating unit and a_i, b_i and c_i are its cost coefficients. P_i is the real power output (MW) of i^{th}generator corresponding to time period t. N_G is the number of online generating units to be dispatched. The cost function for unit with valve point loading effect is calculated by using

$$f_i(P_i) = a_i P_i^2 + b_i P_i + c_i + \left| e_i \sin\left(f_i\left(P_i^{\min} - P_i\right)\right)\right| \tag{3}$$

Where e_i and f_i are the cost coefficients corresponding to valve point loading effect.

This constrained ELD problem is subjected to a variety of constraints depending upon assumptions and practical implications. These include power balance constraints to take into account the energy balance; ramp rate limits to incorporate dynamic nature of ELD problem and prohibited operating zones. These constraints are discussed as under.

Power Balance Constraints or Demand Constraints. This constraint is based on the principle of equilibrium between total system generation ($\sum_{i=1}^{N_G} P_i$) and total system loads (P_D) and losses (P_L). That is,

$$\sum_{i=1}^{N_G} P_i = P_D + P_L \tag{4}$$

where the transmission loss P_L is expressed using B- coefficients, given by

$$P_L = \sum_{i=1}^{N_G} \sum_{j=1}^{N_G} P_i B_{ij} P_j + \sum_{i=1}^{N_G} B_{0i} P_i + B_{00} \tag{5}$$

The Generator Constraints. The power generated by each generator shall be within their lower limit $P_i^{min}$and upper limit P_i^{max}. So that

$$P_i^{\min} \le P_i \le P_i^{\max} \tag{6}$$

The Ramp Rate Limits. Under practical circumstances ramp rate limit restricts the operating range of all the online units for adjusting the generator operation between two operating periods. The generation may increase or decrease with corresponding upper and downward ramp rate limits. So, units are constrained due to these ramp rate limits as mentioned below.

If power generation increases, $P_i - P_i^{t-1} \le UR_i$ (7)

If power generation decreases, $P_i^{t-1} - P_i \le DR_i$ (8)

where P_i^{t-1} is the power generation of unit *i* at previous hour and UR_i and DR_i are the upper and lower ramp rate limits respectively. The inclusion of ramp rate limits modifies the generator operation constraints (6) as follows.

$$\max(P_i^{\min}, UR_i - P_i) \le P_i \le \min(P_i^{\max}, P_i^{t-1} - DR_i) \tag{9}$$

Prohibited Operating Zone. The prohibited operating zones are the range of output power of a generator where the operation causes undue vibration of the turbine shaft. Normally operation is avoided in such regions. Hence, mathematically the feasible operating zones of unit can be described as follows:

$$P_i \le \breve{P}^{pz} \text{ and } P_i \ge \hat{P}^{pz} \tag{10}$$

where $\breve{P}^{pz}$ and $\hat{P}^{pz}$ are the lower and upper are limits of a given prohibited zone for generating unit *i*.

Conflicts in Constraints Handling. If ever, the maximum or minimum limits of generation of a unit as given by (9) lie in the prohibited zone for that generator, then some modifications are to be made in the upper and lower limits for the generator constraints in order to avoid the conflicts. In case, maximum limit for a generator lies in the prohibited zone, the lower limit of the prohibited zone is taken as the maximum limit of power generation for that particular generator. Similarly, care is taken in case the minimum limit of power generation of a generator lies in the prohibited zone by taking upper limit of the prohibited zone as the lower limit of power generation for that generator.

2.2 ELD Formulation with Non-smooth Cost Function

If multiple fuels are used the objective function of an ELD problem is a non-smooth cost function. With multiple fuels, the objective function is a superposition of piecewise quadratic functions

$$F_i(P_i) = \begin{cases} a_{i1} + b_{i1}P_i + c_{i1}P_i^2 & \text{if } P_i^{\min} \le P_i \le P_{i1} \\ a_{i2} + b_{i2}P_i + c_{i2}P_i^2 & \text{if } P_{i1} \le P_i \le P_{i2} \\ . & \\ . & \\ a_{in} + b_{in}P_i + c_{in}P_i^2 & \text{if } P_{in-1} \le P_i \le P_i^{\max} \end{cases} \tag{11}$$

where a_{if}, b_{if}, c_{if} are the cost coefficients of generator 'i' for the 'f_{th}' fuel so that the total cost in this case is given by,

$$F_t^{\cos t} = \sum f_i(P_i) \qquad \text{where } 1 \le i \le N_G \tag{12}$$

3 An Overview of PSO Algorithm

The classical PSO proposed by Kennedy and Eberhart [8] starts with the random initialization of a population of candidate solutions (particles) over the fitness landscape. It works depending on the social behavior of the particles in the swarm. Therefore, it finds the global best solution by simply adjusting the trajectory of each individual towards its own best position and toward the best particle of the entire swarm at each time-step (generation). In a D-dimensional search space, the position vector of the i-th particle is given by $\vec{X}_i = (x_i^1, x_i^2, \cdots, x_i^D)$ and velocity of the i-th particle is given by $\vec{V}_i = (v_i^1, v_i^2, \cdots, v_i^D)$. Positions and velocities are adjusted and the objective function to be optimized $f(\vec{X}_i)$ is evaluated with the new coordinates at each time-step. The velocity and position update equations for the d-th dimension of the i-th particle in the swarm may be represented as:

$$v_i^d = \omega * v_i^d + c_1 * rand1_i^d * (pbest_i^d - x_i^d) + c_2 * rand2_i^d * (gbest^d - x_i^d), \tag{13}$$

$$x_i^d = x_i^d + v_i^d, \tag{14}$$

where c_1 and c_2 are the acceleration constants, c_1 controls the effect of the personal best position, c_2 determines the effect of the best position found so far by any of the particles, $rand1_i^d$ and $rand2_i^d$ are two uniformly distributed random numbers in the range [0, 1]. ω is the inertia weight that balances between the global and local search abilities and takes care of the influence of the previous velocity vector. $pbest_i = (pbest_i^1, pbest_i^2, \cdots, pbest_i^D)$ is the best previous position yielding the best fitness value $pbest_i$ for the i^{th} particle and $gbest = (gbest^1, gbest^2, \cdots, gbest^D)$ is the best position discovered by the whole population. In the local version of PSO, each particle's velocity is modified according to its personal best and the best performance achieved so far within its neighborhood instead of learning from the personal best and the best position achieved so far by the whole population in the global version. The velocity updating equation becomes:

$$v_i^d = \omega * v_i^d + c_1 * rand1_i^d * (pbest_i^d - x_i^d) + c_2 * rand2_i^d * (lbest_i^d - x_i^d), \quad (15)$$

here $lbest_i = (lbest_i^1, lbest_i^2, \cdots, lbest_i^D)$ is the best position achieved within its neighborhood.

4 Lbest-PSO with Dynamically Varying Sub-swarms

The Lbest-Particle Swarm Optimization with Dynamically Varying Sub-swarms (LPSO-DVS) is a variant of PSO constructed based on the local version of PSO. A special neighborhood topology is used in this variant. Similar kind of topology has been used in DMSPSO [9]. As PSO with smaller neighborhood produces better result LPSO-DVS uses smaller neighborhood. As a result the convergence velocity of the population decreases, diversity increases and better solutions are achieved. The total population is divided into a number of small sized swarms whose own members search for better solution in the search space. In order to slow down the population's convergence velocity and increase diversity small neighborhoods are used in LPSO-DVS. Each small sized swarm uses its own members to search for better area in the search space. Since the small sized swarms are searching using their own best historical information, they are easy to converge to a local optimum because of PSO's convergence property. In order to avoid it we vary the no of particles in each subgroup dynamically to keep co-ordination in the search process of individual swarms. After certain number of iterations new particles are added to a sub-swarm and some of the previous particles go to different sub-swarm and thus the neighborhood of a particle changes dynamically. The sizes of the sub-swarms vary in each generation. The number of particles in a sub-swarm may vary from three to ten. The process of variation of sub-swarm is done randomly after 25 iterations (determined empirically, larger no of iterations does not improve the result much but only increases the complexity). In this way, the information obtained by each swarm is exchanged among the swarms. Particles from different swarms are grouped in a new configuration so that each small swarms search space is enlarged and better solutions are possible to be found by the new small swarms. In the velocity update equation we have used a constriction factor to avoid the unlimited growth of the particles' velocity. This was proposed by Clerc and Kennedy [10]. Equation 14 becomes

$$V_i^d = \chi * (\omega * V_i^d + c_1 * rand1_i^d * (pbest_i^d - X_i^d) + c_2 * rand2_i^d * (lbest_i^d - X_i^d)) \quad (16)$$

χ is the constriction factor given by

$$\chi = 2 / \left| 2 - c - \sqrt{c^2 - 4c} \right| \quad (17)$$

$$c = \sum_i c_i \quad (18)$$

5 Results and Discussions

The applicability and viability of the aforementioned technique for practical applications has been tested on three different power systems consisting of different number of units (6, 15 and 40 units). The obtained results are compared with the reported result of other algorithms. The software has been written in MATLAB-7.5 language and executed on a 2 GHz Intel core 2 duo personal computer with 1024-MB RAM. The parameters used for LPSO-DVS are $c_1 = c_2 = 1.49455, \omega = \chi = 0.729$ The results of APSO, HS_PSO, Lbest PSO and LPSO_DVS have been taken from fifty individual runs. The results of the other algorithms are taken from the references.

5.1 Six Unit System

The system contains six thermal generating units. The total load demand on the system is 1263 MW. The results are compared with the GA [11], PSO [12], NPSO-LRS [13], CPSO [14], BBO [15], SOHPSO [16], ABF-NM [17] methods.

Table 1. Comparison of results for 6 unit system

ALGORITHM	Minimum Cost ($/hr.)	Average Cost ($/hr.)
GA	15459	15469
PSO	15450	15454
CPSO1	15447	15449
CPSO2	15446	15449
NPSO_LRS	15450	15450.5
ABF-NM	15443.8164	15446.95383
SOHPSO	15446.02	15497.35
APSO	15443.5751	15449.99
HS_PSO	15444.5734	15454.0819
BBO	15443.0963	15443.0964
Lbest PSO	15443.4815	15444.1953
LPSO_DVS	**15442.6901**	**15443.0311**

5.2 Fifteen Unit System

The system contains fifteen thermal generating units. The total load demand on the system is 2630 MW. The results are compared with the GA [11], PSO [12], CPSO [14], ABF-NM [17] methods for this test system.

Table 2. Comparison of results for 15 unit system

ALGORITHM	Minimum Cost ($/hr.)	Average Cost ($/hr.)
GA	33113	-
PSO	32858	-
CPSO1	32835	33021
CPSO2	32834	33021
ABF-NM	32784.50	32 976.81
APSO	32742.7774	32976.6812
HS_PSO	32692.85714	32740.1885
Lbest PSO	32742.779594	32756.9912
LPSO_DVS	**32692.643046**	**32696.164007**

5.3 Forty Unit System

The system contains forty thermal generating units. The total load demand on the system is 10500 MW. The results are compared with the CEP, FEP, MFEP [18][19], FIA [20], SPSO [21], QPSO [22], and BBO [15] methods for this test system.

Table 3. Comparison of results for 40 unit system

ALGORITHM	Minimum Cost ($/hr.)	Average Cost ($/hr.)
CEP	123488.29	124119.37
FEP	122679.71	124119.37
MFEP	122647.57	123489.74
IFEP	122624.35	123382.00
FIA	121823.80	122662.48
SPSO	121787.39	122474.40
QPSO	121448.21	122225.07
BBO	121426.95	121508.0325
Lbest PSO	125191.40	125519.6954
LPSO_DVS	**121414.84**	**121450.3898**

6 Conclusion

This paper has proposed a new type of PSO algorithm, LPSO_DVS based on dividing the total swarm into sub-swarms and dynamically varying the members of the sub-swarms while finding the local bests and this algorithm is applied to solve constrained economic load dispatch problems. Several non-linear characteristics like ramp rate limits, prohibited operating zones and multiple fuels are also used for practical generator operations. This proposed algorithm has produced results that are better than those generated by other algorithms in the cases discussed; also the convergence characteristics are good. This algorithm has produced better results for both smooth and non smooth cost functions than the other algorithms reported yet. From this limited comparative study, it can be concluded that LPSO_DVS can effectively be used to solve smooth as well as non smooth constrained ELD problems.

References

1. Wood, A.J., Wollenberg, B.F.: Power generation, operation and control. John Wiley & Sons, New York (1984)
2. Chen, C.L., Wang, S.C.: Branch and bound scheduling for thermal generating units. IEEE Trans. Energy Convers 8(2), 184–189 (1993)
3. Yang, H.T., Yang, P.C., Huang, C.L.: Evolutionary Programming based economic dispatch for units with non-smooth fuel cost functions. IEEE Trans. Power Syst. 11(1), 112–118 (1996)
4. Wong, K.P., Fung, C.C.: Simulated annealing based economic dispatch algorithm. Proc. Inst. Elect. Eng. C., Gen., Transm., Distrib. 140(6), 505–519 (1993)
5. Walter, D.C., Sheble, G.B.: Genetic algorithm solution of economic dispatch with valve-point loading. IEEE Trans. Power Syst. 8(3), 1125–1132 (1993)

6. Coelho, L.S., Mariani, V.C.: Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect. IEEE Trans. Power Syst. 21(2), 989–996 (2006)
7. Park, J.B., Lee, K.S., Shin, J.R., Lee, K.Y.: Aparticle swarm optimization for economic dispatch with non-smooth cost functions. IEEE Trans. Power Syst. 20(1), 34–42 (2005)
8. Eberhart, R.C., Kennedy, J.: A new optimizer using particle swarm theory. In: Proc. of the Sixth Int. Symposium on Micromachine and Human Science, Nagoya, Japan, pp. 39–43 (1995)
9. Liang, J.J., Suganthan, P.N.: Dynamic multi-swarm particle swarmoptimizer. In: Proc. Swarm Intell. Symp., pp. 124–129 (June 2005)
10. Clerc, M., Kennedy, J.: The particle swarm–explosion, stability, and convergence in a multidimensional complex space. IEEE Trans. Evol. Comput. 6(1), 58–73 (2002)
11. Walter, D.C., Sheble, G.B.: Genetic algorithm solution of economic dispatch with valve-point loading. IEEE Trans. Power Syst. 8(3), 1125–1132 (1993)
12. Park, J.B., Lee, K.S., Shin, J.R., Lee, K.Y.: A particle swarm optimization for economic dispatch with non-smooth cost functions. IEEE Trans. Power Syst. 20(1), 34–42 (2005)
13. Immanuel Selvakumar, A., Thanushkodi, K.: A new particle swarm optimization solution to nonconvex economic dispatch problems. IEEE Trans. Power Syst. 22(1), 42–51 (2007)
14. Jiejin, C., Xiaoqian, M., Lixiang, L., Haipeng, P.: Chaotic particle swarm optimization for economic dispatch considering the generator constraints. Energy Convers Manage. 48, 645–653 (2007)
15. Bhattacharya, A., Chattopadhyay, P.K.: Biogeography-Based optimization for different economic load dispatch problems. IEEE Trans. Power Syst. 25(2), 1064–1077 (2010)
16. Chaturvedi, K.T., Pandit, M., Srivastava, L.: Self-organizing hierarchical particle swarm optimization for nonconvex economic dispatch problems. IEEE Trans. Power Syst. 23(3), 1079 (2008)
17. Panigrahi, B.K., Ravikumar Pandi, V.: Bacterial foraging optimisation: Nelder–Mead hybrid algorithm for economic load dispatch. IET Gen. Trans. & Distrib. (September 2007)
18. Sinha, N., Chakrabarti, R., Chattopadhyay, P.K.: Evolutionary programming techniques for economic load dispatch. IEEE Trans. Evol. Comput. 7(1), 83–94 (2003)
19. Yao, X., Liu, Y., Lin, G.: Evolutionary programming made faster. IEEE Trans. Evol. Comput. 3(2), 82–102 (1999)
20. Meng, K.: Research of fuzzy self-adaptive immune algorithm and its application, M. E thesis, east China Univ. Sci. Technol., Shanghai, China (2007)
21. Ning, Z.G., Meng, K., Yan, X.F., Qian, F.: An improved particle swarm algorithm and its application in soft sensor modeling. J. East China Univ. Sci. Technol. 33(3), 400–404 (2007)
22. Meng, K., Wang, H.G., Dong, Z.Y.: Quantum-inspired particle swarm optimization for valve-point economic load dispatch. IEEE Trans. Power Syst. 25(1), 215–222 (2010)

Modified Local Neighborhood Based Niching Particle Swarm Optimization for Multimodal Function Optimization

Pradipta Ghosh , Hamim Zafar, and Ankush Mandal

Dept. of Electronics and Telecommunication Engg.,
Jadavpur University, Kolkata 700 032, India
`{iampradiptaghosh,hmm.zafar,ankmd.10}@gmail.com`

Abstract. A particle swarm optimization model for tracking multiple peaks over a multimodal fitness landscape is described here. Multimodal optimization amounts to finding multiple global and local optima (as opposed to a single solution) of a function, so that the user can have a better knowledge about different optimal solutions in the search space. Niching algorithms have the ability to locate and maintain more than one solution to a multi-modal optimization problem. The Particle Swarm Optimization (PSO) has remained an attractive alternative for solving complex and difficult optimization problems since its advent in 1995. However, both experiments and analysis show that the basic PSO algorithms cannot identify different optima, either global or local, and thus are not appropriate for multimodal optimization problems that require the location of multiple optima. In this paper a niching algorithm named as Modified Local Neighborhood Based Niching Particle Swarm Optimization *(ML-NichePSO)*is proposed. The ability, efficiency and usefulness of the proposed method to identify multiple optima are demonstrated using well-known numerical benchmarks.

Keywords: Evolutionary computation, Swarm Intelligence, Multimodal optimization, Niching algorithms, Particle Swarm Optimization, Crowding.

1 Introduction

Multi-objective optimization is one of the leading research fields of modern day. Various algorithms like PSO [1], DE [2], and GA and many more are available for solving single objective optimization problems. But multi-objective problems are not so easy to solve. In case of multi-objective problems the presence of multiple global and local optimal solutions makes it complex. In case of multi-objective problems as there are many peaks and we have to point out all the peaks, it is much more complicated to optimize. Also it will be much better if one algorithm can find all the local optima along with every global optima. However, many practical optimization problems have multiple optima and it is wise to find as many such optima as possible for a number of reasons. Research on solving multimodal problems with EAs dates back to the landmark work of Goldberg and Richardson [3], in which they nicely

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 199–208, 2011.
© Springer-Verlag Berlin Heidelberg 2011

showed the way of introducing niche-preserving technique in a standard Genetic Algorithm (GA) to obtain multiple optimal solutions. Currently the most popular niching techniques used in conjunction with the evolutionary computation community include crowding [4], fitness sharing [3], restricted tournament selection [5], and speciation [6]. Most of existing niching methods, however, possess some difficulties such as: difficulties to pre-specify some niching parameters; difficulties in maintaining discovered solutions in a run; extra computational overhead, and poor scalability when dimensionality is high. The current research on evolutionary multimodal optimization aims at outsmarting these problems by devising more and more efficient optimizers.

In this paper we propose a simple yet very powerful hybrid EA that is based on Local Neighborhood based Particle swarm optimization for multimodal optimization. Since its invention in 1995 by Kennedy and Eberhart, PSO [1] is the base of a huge set of evolutionary algorithm. PSO has emerged as a very competitive optimizer for continuous search spaces.

The reason for employing the PSO in multimodal optimization process is its high exploring power over the bound-constrained search space and also its ability to search locally. The newly proposed ML-NichePSO has the capability to converge more accurately to the local and global peaks, starting from a uniform initialization in the search space, in comparison to a number of state-of-the-art algorithms for multimodal optimization. We have empirically demonstrated that the performance of the algorithm is insensitive to this parameter provided its values are selected as described later. Empirical formulae for these bounds have also been provided and validated. Finally we have demonstrated the efficacy of the proposed algorithm on a set of multimodal optimization problems that constitute the benchmark.

2 Niching Related Works

2.1 Crowding and Restricted Tournament Selection

Crowding introduced by De Jong [7] allows only a fraction of the population to reproduce and die in each generation. It encourages competition for limited resources among similar individuals in the population. Belonging to this category are crowding, deterministic crowding restricted tournament selection (RTS), and so on.

2.2 Sharing and Clustering

Fitness sharing [8] is based on the concept that a point in a search space has limited resources that need to be shared by any individuals that occupy similar search space behaviors or genetic representations. Sharing in EAs is implemented by scaling the fitness of an individual based on the number of “similar” individuals present in the population.

2.3 Clearing

Clearing (CLR) is an alternative to the sharing methods. Instead of sharing resources between all individuals of a niche as in the fitness sharing scheme, clearing attributes them only to the best members of each niche and eliminates the other individuals within the same niche.

2.4 Speciation

The concept of speciation depends on radius parameter r_s, which measures Euclidean distance from the center of a species to its boundary. All individuals falling within the radius from the species seed are identified as the same species.

In addition to the methods listed above, there are niching methods such as sequential niching including de-rating, parallelization, and clustering and others.

3 Overview of the Proposed ML-NichePSO Algorithm

The classical PSO proposed by Kennedy and Eberhart [1] depends on the social interaction between independent agents, here called particles, during their search for the optimum solution using the concept of fitness. After defining the solution space and the fitness function, the PSO algorithm starts by randomly initializing the position and velocity of each particle in the swarm. Each particle in PSO has an adaptable velocity. Moreover, each particle has a memory remembering the best position of the search space that has ever been visited. The velocity matrix is updated according to

$$v_{mn}^{t} = wv_{mn}^{t-1} + c_1 U_{n1}^{t}(p_{mn}^{t} - x_{mn}^{t-1}) + c_2 U_{n2}^{t}(g_{n}^{t} - x_{mn}^{t-1}) \tag{1}$$

where the superscripts *t* and *t-1* refer to the time index of the current and the previous iterations,U_{n1} and U_{n2} are two uniformly distributed random numbers in the interval [0,1]. The position matrix is updated each iteration according to

$$X^{t} = X^{t-1} + V^{t} \tag{2}$$

In the local version of PSO each particle's velocity is adjusted according to its personal best position and the best performance achieved so far within its neighborhood. Then the velocity updating equation is modified as follows.

$$v_{mn}^{t} = wv_{mn}^{t-1} + c_1 U_{n1}^{t}(p_{mn}^{t} - x_{mn}^{t-1}) + c_2 U_{n2}^{t}(l_{mn}^{t} - x_{mn}^{t-1}) \tag{3}$$

Here l_{mn}^{t} is the best position achieved within its neighborhood after *t* iterations.

3.1 Description of the Proposed Algorithm

This algorithm is a modified version of PSO with local neighborhood but applied as niching technique. The neighborhood of each particle is selected dynamically (i.e. it can change in every iteration). A particle is selected as neighbor of another particle and vice versa if they satisfy certain conditions.

1. The first condition is that the distance between them must be less than or equal to some predefined maximum distance ($D_{\max}$). The Maximum distance is generally equal to one third of the distance between two nearest optima found by the particles till the present iteration. All the particles within a circle or radius $D_{\max}$ with centre at the position of the particle, whose neighborhood has to be defined, are selected as neighbors of that particle. But initial value of $D_{\max}$ is selected using a generalized equation for all dimensions given below.

$$D_{\max_{initial}} = \frac{\sqrt{\sum_{k=1}^{D} (x_k^u - x_k^l)^2}}{10 * Dim} \tag{4}$$

where x_k^l and x_k^u are the lower and upper bounds on the k^{th} dimension of the variable vector of *Dim* dimensions. This empirical equation gives us much better results than the results without any initial definition of $D_{\max}$. The value of $D_{\max}$ is updated in the rest of the iterations as follows.

$$D_{\max} = \text{Min (Distances between two optima found so far)/3;} \tag{5}$$

2. Another selecting criterion is involved in this algorithm. This method provides the dynamic nature of neighborhood. A random number is selected between 1 and 0. If it is 1for a neighborhood's particle in a particular iteration, then it is the active neighbor of the selected particle for that iteration. Otherwise, the neighbor particle is simply ignored.

The grouping of particles is clearly shown in Figure 2. In every iteration each particle compares its local best (***lbest)*** position with its neighbor's ***lbest*** positions and updates its velocity and position using equations 2 and 3 if the above two conditions are met. This updating of velocity and position is performed with probability of 75%. This probability is considered for much more variation in the new particle's position and velocity, also it can create new particles similar to the crossover vector in DE terminology. Figure 1 presents a simplified block diagram of our proposed algorithm.

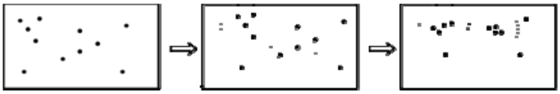


Fig. 1. Block diagram of the ML-NichePSO algorithm

Still there are several problems to be solved. In our algorithm we have provided solutions of these problems. They are listed below.

A. Some of the particles may move outside the search space during searching operation. In our algorithm, in order to constrain the particles within the search space, we calculate the fitness value of a particle and update its *position* only if the particle is within the search space. If a particle moves out of the range it is reinitialized. This feature is not shown explicitly in pseudo code.

B. There is always a possibility of crowding of large number of particles in any of the optima. In case of all dimensions, especially high dimensions (Where Dimension is ≥ 10), there is always a high probability of gathering a large proportion of the population in same peak. Due to this reason the probability of finding most of the peaks is reduced. To eradicate this problem in our algorithm we have provided a check whether more than 3 particles are very close to each other or not. We have incorporated a scattering scheme to improve the searching ability. Whenever the aforesaid phenomenon i.e. crowding of a number of particles at the same position occurs this scheme scatters most of the particles to a distant position leaving only few at that location. The scattering of particles can be compared with the physical phenomenon observed in ants when we put something in the location of their gathering except that in our algorithm some particles are remained at the position of gathering. This feature helps to repel particles which are very close or almost same in position and thus helps to increase the probability of finding more peaks as particles are distributed in the search space to find new peaks. As a result of this scattering process the previously found peak is not lost as at least one particle is retained in that position.

The scattering is done via following steps.

Step1. Check the no of particles located at same position. Two particles will be considered to be at same position iff they satisfy a particular distance criterion. If the distance between two particles is less than 0.5% of $D_{\max}$, they will be considered to be at the same position.

Step2. Put aside the best 3 particles in terms of fitness for that particular location.

Step3. Rest of the particles' locations are checked dimension wise i.e. whether they are on the lower bound side or upper bound side of the best particle in that location for that particular dimension.

Step4. Now the scattering is incorporated using the following formulas.

If the particle is on the lower bound side (i.e. within lower bound and best position) for i^{th} dimension

$$Position^{i}_{updated} = Position^{i}_{best} - (Position^{i}_{best} - lower\_bound^{i}) \times (0.5) \times rand() \tag{6}$$

If the particle is on the upper bound side (i.e. within upper bound and best position) for i^{th} dimension

$$Position^{i}_{updated} = Position^{i}_{best} + (upper\_bound^{i} - Position^{i}_{best}) \times (0.5) \times rand() \tag{7}$$

4 Experiments and Results

4.1 Numerical Benchmarks

The collection of the test functions are mainly composed of 8 multimodal general functions and 7 composition functions of various characteristics, such as irregular landscape, symmetric or equal distribution of optima, unevenly spaced optima, etc. A brief description of the functions is given in Table 2.

4.2 Algorithms Compared

Performance of ML-NichePSO is compared with the following standard multimodal evolutionary algorithms:

Crowding DE (CDE) [3] , Speciation-based DE (SDE) [8], Fitness-Euclidean distance Ratio PSO (FER-PSO) [9], Speciation-based PSO (SPSO) [10], r2pso [11], r3pso [11], r2psolhc [11], r2psolhc [11], and CMA-ES with Self-Adaptive Niche Radius (S-CMA).

4.3 Performance Measure

The performance of all multimodal algorithms is measured in terms of the following two criteria:

1) **Success Rate:** The percentage of runs in which all global peaks are successfully found. Table 3 shows the respective data.
2) **Average Number of optima found:** This is the most important performance measuring criteria of any niching algorithm.

All performances are calculated and averaged over 50 independent runs. All the algorithms are implemented in MATLAB 7.5 and executed using a Pentium core 2 duo machine with 2 GB RAM and 2.23 GHz speed.

4.4 Numerical Results

This section presents the experimental results. All the algorithms are run till the maximum number of FEs are exhausted. The second and third column of tables 3 and 4 indicate the level of accuracy and niche radius (for SDE and SPSO) used in the experiments. Values of niche radius values are chosen as recommended by the corresponding authors. The level of accuracy(ε) is taken too small so that it becomes challenging for the algorithms to locate the peaks. For $f_{7,}$ f_8 and all the 10-D composition functions, since it is very difficult to achieve a non-zero success rate, the comparison is based only on the average number of optima found.

4.5 Population and Max_FEs Initialization

Table 1. Initialization of population

Function Number	Population Size	Maximum number of FEs
f_1 to f_6	50	10000
f_7	100	20000
f_8	200	40000
*CF*1 to *CF*7	100	50000

Table 2. Benchmark Functions

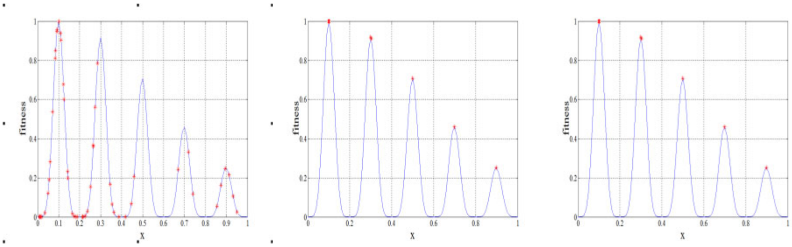
Name	Dim	Test Function	Range	No. of global peaks
f_1:Decreasing maxima	1	$f_5(x)=\exp\left[-2\log(2)\cdot(\frac{x-0.1}{0.8})^2\right]\cdot\sin^6(5\pi x)$	$0\le x\le 1$	1
f_2:Uneven maxima	1	$f_2(x)=\sin^6(5\pi(x^{\frac{3}{4}}-0.05))$	$0\le x\le 1$	5
f_3: Uneven Decreasing Maxima	1	$f_3(x)=\exp\left[-2\log(2)\cdot(\frac{x-0.08}{0.854})^2\right]\cdot\sin^6(5\pi(x^{\frac{3}{4}}-0.05))$	$0\le x\le 1$	1
f_4: Himmelblau's function	2	$f_4(\vec{X})=200-(x_1^2+x_2-11)^2-(x_1^2+{x_2}^2-7)^2$	$-4\le x_1,x_2\le 4$	4
f_5: Six-hump camel back	2	$f_5(\vec{X})=-4\left[\left(4=2.2x_1^2+\frac{x_1^4}{3}\right)x_1^2+x_1x_2+(-4+4x_2^2)x_2^2\right]$	$-1.9\le x_1\le 1.9$ $-1.1\le x_2\le 1.1$	2
f_6: 1D inverted Vincent function	1	$f(\vec{X})=\frac{1}{n}\sum_{i=1}^{n}\sin(10\cdot\log(x_i)),$ Where n is the dimensionality of the problem.	$0.25\le x_i\le 10$	6
f_7: 2D inverted Vincent function	2			36
f_8: 3D inverted Vincent function	3			216
CF1	10	Corresponds to CF1 of [12]	$-5\le x_i\le 5$	8
CF2	10	Corresponds to CF2 of [12]	$-5\le x_i\le 5$	6
CF3	10	Corresponds to CF3 of [12]	$-5\le x_i\le 5$	6
CF4	10	Corresponds to CF4 of [12]	$-5\le x_i\le 5$	6
CF5	10	Corresponds to CF5 of [12]	$-5\le x_i\le 5$	6
CF6	10	Corresponds to CF6 of [12]	$-5\le x_i\le 5$	6
CF7	10	Corresponds to CF9 of [12]	$-5\le x_i\le 5$	6

Table 3. Success Rates for test functions

Function	ε	r	ML-NichePSO	CDE	SDE	CMA-ES	FERPSO	SPSO	r2pso	r3pso	r2psolhc	r3psolhc
f_1	0.000001	0.01	**100**	72	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
f_2	0.000001	0.01	**100**	28	60	88	**100**	92	88	72	92	92
f_3	0.000001	0.01	**100**	60	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
f_4	0.0005	0.5	**100**	0	72	72	72	0	28	24	28	24
f_5	0.000001	0.5	**100**	0	**100**	60	96	0	56	60	56	52
f_6	0.0001	0.2	**100**	56	48	60	60	72	68	56	52	48

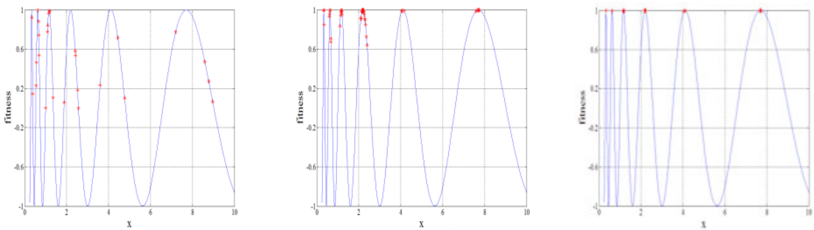
Table 4. Average number of peaks found for the test functions

Function	ε	r	ML-NichePSO	CDE	SDE	CMA-ES	FERPSO	SPSO	r2pso	r3pso	r2psolhc	r3psolhc
f_1	0.000001	0.01	**1**	0.72	**1**	1	**1**	**1**	**1**	**1**	**1**	**1**
f_2	0.000001	0.01	**5**	3.96	4.6	4.88	**5**	4.92	4.88	4.72	4.92	4.88
f_3	0.000001	0.01	**1**	0.6	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
f_4	0.0005	0.5	**4**	0.32	3.72	3.72	3.68	0.84	2.92	2.76	3	3.12
f_5	0.00001	0.5	**2**	0.04	**2**	1.6	1.96	0.08	1.44	1.56	1.56	1.48
f_6	0.0001	0.2	**6**	5.56	4.88	5.56	5.36	5.6	5.52	5.16	5.36	5.28
f_7	0.001	0.2	30.6	**33.8**	22.8	24.6	23.6	25.72	21.8	22.2	22.52	23.12
f_8	0.001	0.2	101	**152**	50.6	75.6	68.6	70.12	40.6	45.4	42.2	43.32
CF1	0.5	1	**2**	0	1.8	1.08	1.08	0	0	0	0	0
CF2	0.5	1	**2**	1.2	1.2	1.52	**2**	0	0	0	0	0
CF3	0.5	1	2	0.72	1.52	0.52	**2.52**	0	0	0	0	0
CF4	0.5	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
CF5	0.5	1	**2**	1.12	1.32	1.2	**2**	0	0	0	0	0
CF6	0.5	1	**2**	0	1.4	1.12	1.2	0	0	0	0	0
CF7	0.5	1	1	0	**1.8**	1.2	1.52	0	0	0	0	0



a)FEs=100 b) FEs=500 c) FEs=1000

Fig. 2. Distribution of individual particles in the search space during the evolution of the process for f_1

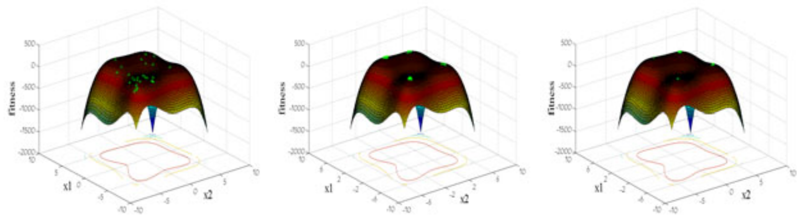


a)FEs=100 b)FEs=500 c) FEs=1000

Fig. 3. Distribution of individuals in the search space during the evolution of the process for f_6

4.6 Analysis of Results

The results shown above clearly indicate it has the capability to locate the local peaks satisfactorily other than the global peaks, which should be the target of a multimodal optimization algorithm. In addition, it has also performed efficiently when the



a)FEs=500 b)FEs=1000 c) FEs=5000

Fig. 4. Distribution of individuals in the search space during the evolution of the process for f_4

dimensionality is high (D =10). For these high-dimensional composite functions, the traditional PSO variants have completely failed to locate any of the global optima. The algorithm offers a fast convergence towards the results as proved by the figures where almost all the peaks are found in 500FEs only. For two dimensional problems, all the peaks are found after 1000 FEs and after 5000 FEs all the particles are located in any of the global peaks. So this algorithm is time saving, less complex, and efficient in finding the global peaks.

5 Conclusions

In this paper we proposed a multimodal evolutionary optimization technique that integrates concepts of Local Search based Particle Swarm Optimization. Its performance over fifteen Multimodal Benchmark functions are compared with nine state-of-the-art evolutionary multi-modal optimizers based on the performance metrics like success rate, average number of peaks found and peak accuracy. From the Analysis of the results we can firmly propose our algorithm as one of the most powerful niching algorithm present in various publications. Our further work will be to improve our proposed algorithm to use it effectively and efficiently in tracking multiple peaks in a dynamic environment.

References

1. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proc. IEEE Conf. Neural Networks IV, Piscataway, NJ (1995)
2. Storn, R., Price, K.: Differential evolution a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optimization 11(4), 341–359 (1997)
3. Goldberg, D.E., Richardson, J.: Genetic algorithms with sharing for multimodal function optimization. In: Proceedings of the Second International Conference on Genetic Algorithms, pp. 41–49 (1987)
4. Thomsen, R.: Multimodal optimization using Crowding-based differential evolution. In: Proceedings of the IEEE 2004 Congress on Evolutionary Computation, pp. 1382–1389 (2004)
5. Harik, G.R.: Finding multimodal solutions using restricted tournament selection. In: Proceedings of the 6th International Conference on Genetic Algorithms, San Francisco, pp. 24–31 (1995)

6. Petrowski, A.: A clearing procedure as a niching method for genetic algorithms. In: Proc. of 3rd IEEE Congress on Evolutionary Computation, pp. 798–803 (1996)
7. De Jong, K.A.: An analysis of the behavior of a class of genetic adaptive systems, Doctoral Dissertation, University of Michigan (1975)
8. Kennedy, J.: Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance. In: Proc. of IEEE Congress on Evolutionary Computation (CEC 1999), Piscataway, NJ, pp. 1931–1938 (1999)
9. Li, X.: Multimodal function optimization based on fitness-euclidean distance ratio. In: Proc. Genet. Evol. Comput. Conf., pp. 78–85 (2007)
10. Li, X.: Adaptively Choosing Neighbourhood Bests Using Species in a Particle Swarm Optimizer for Multimodal Function Optimization. In: Deb, K., et al. (eds.) GECCO 2004. LNCS, vol. 3102, pp. 105–116. Springer, Heidelberg (2004)
11. Li, X.: Niching without niching parameters: particle swarm optimization using a ring topology. IEEE Transaction on Evolutionary Computation 14(1) (February 2010)
12. Qu, B.Y., Suganthan, P.N.: Novel multimodal problems and differential evolution with ensemble of restricted tournament selection. In: IEEE Congress on Evolutionary Computation, Barcelona, Spain (July 2010)
13. Qu, B.Y., Suganthan, P.N., Liang, J.J.: Differential Evolution with Neighborhood Mutation for Multimodal Optimization. IEEE Trans. on Evolutionary Computation, doi:10.1109/TEVC.2011.2161873
14. Das, S., Maity, S., Qu, B.-Y., Suganthan, P.N.: Real-parameter evolutionary multimodal optimization — A survey of the state-of-the-art. Swarm and Evolutionary Computation 1(2), 71–88 (2011)

Constrained Function Optimization Using PSO with Polynomial Mutation

Tapas Si[1], Nanda Dulal Jana[1], and Jaya Sil[2]

[1] Department of Information Technology, National Institute of Technology, Durgapur, West Bengal, India
[2] Department of Computer Science and Technology, BESU, West Bengal, India
`{c2.tapas,nanda.jana}@gmail.com`
`js@cs.becs.ac.in`

Abstract. Constrained function optimization using particle swarm optimization (PSO) with polynomial mutation is proposed in this work. In this method non-stationary penalty function approach is adopted and polynomial mutation is performed on global best solution in PSO. The proposed method is applied on 6 benchmark problems and obtained results are compared with the results obtained from basic PSO. The experimental results show the efficiency and effectiveness of the method.

1 Introduction

Constrained Optimization (CO) problems are encountered in numerous applications. Structural optimization, engineering design, VLSI design, economics, allocation and location problems are just few of the scientific fields in which CO problems are frequently met [1]. The CO problem can be represented as the following nonlinear programming problem:

$$minimize\ f(X), X = x_1, x_2, \dots, x_n \tag{1}$$

Subject to:

$$g_i(X) \le 0, i = 1,2, \dots, p \tag{2}$$

$$h_j(X) = 0, j = p + 1, \dots, m \tag{3}$$

Usually equality constraints are transformed into inequalities of the form

$$\left| h_j\,(X) - \varepsilon \right| \le 0\,, for\ j = p + 1, \dots, m \tag{4}$$

A solution X is regarded as feasible if $g_i(X) <= 0$ for i = 1,2,…,p and $|h_j(X) - \varepsilon| = 0$ for j =p +1,….,m. In this work, we used $\varepsilon = 0.0001$.

The most common approach for solving CO problems is the use of a penalty function. The constrained problem is transformed to an unconstrained one, by penalizing the constraints and building a single objective function, which in turn is minimized using an

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 209–216, 2011.
© Springer-Verlag Berlin Heidelberg 2011

unconstrained optimization algorithm. The CO problem is tackled through the minimization of a non-stationary multistage assignment penalty function [1].

Different mutation strategies like Cauchy Mutation [3], Gaussian Mutation [2],Power Mutation [6], Adaptive Mutation [5] etc. are introduced into PSO for solving unconstrained function optimization problems. In this work, our objective is to use polynomial mutation [9] on global best solution in PSO to solve constrained function optimization problems. The results of experiments performed on well-known test problems are reported and discussed in comparison with results obtained by basic PSO.

2 Non-stationary Penalty Function

The Penalty Function technique solves the CO problem through a sequence of unconstrained optimization problems. Up to date, no other method for defining pertinent functions, than trial-and-error, exists. If the penalty values are high, the minimization algorithms usually get trapped in the local minima. On the other hand, if penalty values are too low, they can hardly detect feasible optimal solutions. Penalty functions are distinguished into two main categories: stationary and non-stationary. Stationary penalty functions use fixed penalty values throughout the minimization, while in contrast, in non-stationary penalty functions, the penalty values are dynamically modified. A penalty function is generally defined as

$$F(x) = f(x) + h(k)H(x), x \epsilon S \subset R^n \tag{5}$$

where f(x) is the original objective function of the CO problem in Eq. (1); h(k) is a dynamically modified penalty value, where k is the algorithms current iteration number; and H(x) is a penalty factor, defined as

$$H(x) = \sum_{i=1}^{m} \theta(q_i(x))\, q_i(x)^{\gamma(q_i(x))} \tag{6}$$

where $q_i(x) = \max(0, g_i(x)), i = 1, \dots, m$;The function is a relative violated function of the constraints;$\theta(q_i(x))$ is a multi-stage assignment function;$\gamma(q_i(x))$ is the power of the penalty function; and $g_i(X)$ are the constraints described in Eq. (2). The functions $h(.), \theta(.), \gamma(.)$ are problem dependent. In our experiments, a non-stationary multi-satge assignment penalty function was used.

3 Particle Swarm Optimization (PSO)

PSO [14] is a kind of algorithm to search for the best solution by simulating the movement and flocking birds. PSO algorithms use a population of individual called particles. Each particle has its own position and velocity to move around the search space. Particles have memory and each particle keep track of previous best position and corresponding fitness. The previous best value is called as *pbest*. Thus *pbest* is related only to a particular particle. It also has another value called gbest, which is the best value of all the particles *pbest* in the swarm. The basic concept of PSO technique lies in accelerating each particle towards its *pbest* and the locations at each time step. Global best solution *gbest* is the best of the *pbest* of all particles in the swarm space.

$$V_{ij} = w \times V_{ij} + C1 \times R1 \times \left(X_{pbest} - X_{ij}\right) + C2 \times R2 \times (X_{gbest} - X_{ij}) \tag{7}$$

$$X_{ij} = X_{ij} + V_{ij} \tag{8}$$

$C1$ and C2 are personal and social cognigence of a particles respectively and R1 and R2 are two uniformly distributed random numbers in the interval (0,1).The inertia weight w in (7) was introduced by Shi and Eberhart [14]. They proposed a w linearly decreasing with the iterative generation as

$$w = w_{max} - (w_{max} - w_{min})\frac{g}{G} \tag{9}$$

where g is the generation index representing the current number of evolutionary generation and G is the predefined maximum number of generations.

4 PSO Based on Polynomial Mutation

The particle swarm optimization algorithms converges rapidly during the initial stages of a search, but often slows considerably and can get trapped in local optima. Mutation strategies are used into PSO in the hope of preventing PSO from trapping into a local optimum through long jumps made by the mutation. Polynomial mutation is based on polynomial probability distribution.

$$x_j(t+1) = x_j(t) + \left(x_j^u - x_j^l\right) \times \delta_j \tag{10}$$

where x_j^u is the upper bound and x_j^l is the lower bound of x_j. The parameter δ_j is calculated from the polynomial probability distribution.

$$P(\delta) = 0.5 \times (\eta_m + 1)(1 - \delta^{|\eta_m|}) \tag{11}$$

η_m is the polynomial distribution index.

$$\delta_j = \begin{cases} (2r_j)^{1/(\eta_m+1)} - 1, r_j < 0.5 \\ 1 - 2[(1 - r_j)]^{1/(\eta_m+1)}, r_j \geq 0.5 \end{cases} \tag{12}$$

The property of η_m is such that by varying its value, the perturbance can be varied in the mutated solution. If the value of η_m is large, a small perturbance in the value of a variable is achieved. To achieve gradually decreasing perturbance in the mutated solutions, the value of η_m is gradually increased. The following rule is to achieve the above adaptation which is known as adaptive polynomial mutation:

$$\eta_m = 80 + t \tag{13}$$

where t is the current iteration. In this work, we used polynomial mutation on global best solution in PSO (**PSO-PM**) using the following equation:

$$mgbest_j(t) = gbest_j(t) + \left(x_j^u - x_j^l\right) \times \delta_j \tag{14}$$

where x_j^u is the upper bound and x_j^l is the lower bound of x_j. If mutated global best $mgbest$ is better than $gbest$, then $gbest$ is replaced by $mgbest$.

5 Parameter Settings

5.1 Test Suite

All the Test Problems are obtained from CEC 2010 competition on constrained real parameter optimization [8].

Test Problem 1:

$$Min\ f(x) = \left|\frac{\sum_{i=1}^{D} cos^4(z_i) - 2\prod_{i=1}^{D} cos^2(z_i)}{\sqrt{\sum_{i=1}^{D} iz_i^2}}\right| \quad z = x - o$$

$$g_1(x) = 0.75 - \prod_{i=1}^{D} z_i \le 0$$

$$g_2(x) = \sum_{i=1}^{D} z_i - 7.5D \le 0$$

$$x \epsilon [0,10]^D$$

Test Problem 2:

$$Min\ f(x) = \frac{1}{D}\sum_{i=1}^{D}\left(z_i \sin\left(\sqrt{|z_i|}\right)\right) \quad z = x - o$$

$$h(x) = \sum_{i=1}^{D-1}(z_i^2 - z_{i+1})^2 = 0$$

$$g(x) = \sum_{i=1}^{D}(z_i - 100\cos(0.1z_i) = 10) \le 0$$

$$x \epsilon [-1000,1000]^D$$

Test Problem 3:

$$Min\ f(x) = \frac{1}{D}\sum_{i=1}^{D}\left(-z_i \sin\left(\sqrt{|z_i|}\right)\right) \quad z = x - o$$

$$g_1(x) = -50 + \frac{1}{100D}\sum_{i=1}^{D} {z_i}^2 \le 0$$

$$g_2(x) = \frac{50}{D}\sum_{i=1}^{D} \sin\left(\frac{1}{50}\pi z_i\right) \le 0$$

$$g_3(x) = 75 - 50(\sum_{i=1}^{D} \frac{z_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1) \leq 0$$

$$x\epsilon[-500{,}500]^D$$

Test Problem 4:

$$Min\ f(x) \ = \sum_{i=1}^{D} \frac{z_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1 \qquad z = x - o$$

$$g_1(x) = \sum_{i=1}^{D} [z_i^2 - 100\cos(\pi z_i) + 10] \leq 0$$

$$g_2(x) = \prod_{i=1}^{D} z_i \leq 0$$

$$h_1(x) = \sum_{i=1}^{D} \left(z_i \sin\left(\sqrt{|z_i|}\right)\right) = 0$$

$$h_2(x) = \sum_{i=1}^{D} \left(-z_i \sin\left(\sqrt{|z_i|}\right)\right) \ = 0$$

$$x\epsilon[-10{,}10]^D$$

Test Problem 5:

$$Min\ f(x) = \sum_{i=1}^{D-1} (z_i^2 - z_{i+1})^2 \qquad z = x - o$$

$$g_1(x) = \prod_{i=1}^{D} z_i \leq 0$$

$$g_2(x) = \sum_{i=1}^{D} z_i \leq 0$$

$$h(x) = \sum_{i=1}^{D} \left(z_i \sin\left(4\sqrt{|z_i|}\right)\right) = 0$$

$$x\epsilon[-10{,}10]^D$$

Test Problem 6:

$$Min\ f(x) = \sum_{i=1}^{D-1} (z_i^2 - z_{i+1})^2 \qquad z = x - o$$

$$g(x) = \frac{1}{D}\sum_{i=1}^{D}\left(-z_i \sin\left(\sqrt{|z_i|}\right)\right) \leq 0$$

$$h(x) = \frac{1}{D}\sum_{i=1}^{D}\left(z_i \sin\left(\sqrt{|z_i|}\right)\right) = 0$$

$$x\epsilon[-10,10]^D$$

5.2 Parameters

1. Population Size(N)=100.
2. Number of Generations= FES/*Population size* (where FES is the number of function evolution permitted)
3. In our experiments, we used FES=1x10^5 for 10D problems and FES=3x10^5 for 30D problems.
4. Distribution index for polynomial mutation η_m= 100
5. C1=C2=1.49445
6. W_{max}=0.9, W_{min}=0.4

We adopted the same parameter settings from [1].Specifically, if $q_i(x)$ <1, then, $\gamma(q_i(x))$=1,otherwise $\gamma(q_i(x))$=2.Moreover, if $q_i(x)$ <0.001 ,then $\theta(q_i(x))$=10 ,else, if $q_i(x)$<0.01,then $\theta(q_i(x))$=20,else,if $q_i(x)$<=1,then $\theta(q_i(x))$=100,otherwise $\theta(q_i(x))$=300,Regarding the function h(.), it was set to $h(k) = k\sqrt{k}$ where k is the current iteration number.

6 Simulation Results

The obtained results are presented in Table 1, 2, 3, 4 and 5. Best, mean, standard deviation and sum of violated constraints at the best solution of 25 runs for each test problem are reported in Table 2, 3, 4 and 5. Feasible run is one during which at least one feasible solution is found. The feasibility rate is defined as: (number of feasible runs)/ 25. Feasibility Rates for 10D and 30D problems are reported in Table 1.

Table 1. Feasibility Ratio of 10D and 30D problems

Problem	10D		30D	
	PSO	**PSO-PM**	**PSO**	**PSO-PM**
1	1	1	1	1
2	0	0	0	0
3	0.32	0.28	0	0
4	1	1	1	1
5	1	1	1	1
6	0.96	0.88	0.92	0.96

Table 2. Function values achieved for 10D problems using PSO

Problem	Best	Mean	Std. Dev.	Sum of V.C
1	-0.6601	-0.6041	0.0395	0
2	-2144.64	-93.0875	743.0699	1.80E+13
3	-73.0048	-9.3149	40.9014	290.9384
4	0.51823	0.9148	0.1478	0
5	9.0022	62.9733	38.1994	0
6	12.8234	59.0856	54.4245	0

Table 3. Function values achieved for 10D problems using PSO-PM

Problem	Best	Mean	Std. Dev.	Sum of V.C
1	-0.7007	-0.6086	0.0545	0
2	-1566.16	-18.6784	917.653	1.70E+13
3	-86.8291	-1.8973	42.0647	340.0157
4	0.65006	0.9648	0.0996	0
5	17.5714	57.7369	34.0382	0
6	21.36	86.2254	61.7331	0

Table 4. Function values achieved for 30D problems using PSO

Problem	Best	Mean	Std. Dev.	Sum of V.C
1	-0.5576	-0.436	0.0637	0
2	-3323.7305	-107.4847	1591.6	3.40E+13
3	-24.5271	18.6392	23.1737	564.8598
4	1.037	1.0842	0.0259	0
5	196.3237	373.2904	101.117	0
6	83.8721	202.6687	85.7776	0

Table 5. Function values achieved for 30D problems using PSO-PM

Problem	Best	Mean	Std. Dev.	Sum of V.C
1	-0.564	-0.4537	0.0756	0
2	-3300.7746	-41.3967	1796.7	2.60E+13
3	-68.9134	-2.1241	33.1196	533.8723
4	1.0389	1.0796	0.0324	0
5	162.5888	366.327	138.7617	0
6	104.9394	227.7095	50.6156	0

7 Conclusions

In this work, we proposed polynomial mutation on global best solution in particle swarm optimization and solve constrained optimization function with penalty function approach. We solved the same problems using basic PSO and from the experimental results, it is found that our proposed method performed better than basic PSO. It is also found that effectiveness of mutation on global best solution depends on the search space. If search space is too large and dimension is high for a problem (for test problem 2 and 3), then efficiency of mutation is poor comparatively. In the context of constrained optimization problem, mutation may lead to an unfeasible solution. Therefore, mutation may enforce to direct the particles towards unfeasible region of the search space. An investigation in details of mutation in PSO can be made to solve constrained optimization problems as a future work. As PSO is dependent on different parameters settings, fine-tuning of the parameters may lead to enforce feasibility of the particles and guide swarm towards the optimum solution.

References

1. Parsopoulos, K.E., Vrahatis, M.N.: Particle Swarm Optimization Method for Constrained Optimization Problems. In: ACM Symposium on Applied Computing - SAC (2002)
2. Higashi, N., lba, H.: Particle Swarm Optimization with Gaussian Mutation. In: IEEE Swarm Intelligence Symposium, Indianapolis, pp. 72–79 (2003)
3. Wang, H., Liu,Y., Li,C.H., Zeng, S.Y.: A hybrid particle swarm algorithm with Cauchy mutation. In: Proc. of IEEE Swarm Intelligence Symposium, pp. 356–360 (2007)
4. Stacey, A., Jancic, M., Grundy, I.: Particle swarm optimization with mutation. In: Proc. Congr. Evol. Comput., pp. 1425–1430 (2003)
5. Tang, J., Zhao, X.: Particle Swarm Optimization with Adaptive Mutation. In: WASE International Conference on Information Engineering (2009)
6. Wu, X., Zhong, M.: Particle Swarm Optimization Based on Power Mutation. In: ISECS International Colloquium on Computing, Communication, Control, and Management (2009)
7. Saha, A., Datta, R., Deb, K.: Hybrid Gradient Projection based Genetic Algorithms for Constrained Optimization. In: IEEE Congress on Evolutionary Computation - CEC, pp. 1–8 (2010)
8. Mallipeddi, R., Suganthan, P.: Problem Definitions and Evolution Criteria for the CEC 2010 Competition on Constrained Real-Parameter Optimization, Technical Report, Nayang Technological University, Singapore (2010)
9. Raghuswanshi, M.M., Kakde, O.G.: Survey on multiobjective evolutionary and real code genetic algorithms Complexity. International 11 (2005)
10. Gao, Y., Ren, Z.: Adaptive Particle Swarm Optimization Algorithm With Genetic Mutation Operation. In: Third International Conference on Natural Computation (ICNC 2007) (2007)
11. Li, C., Liu, Y., Zhou, A., Kang, L., Wang, H.: A Fast Particle Swarm Optimization Algorithm with Cauchy Mutation and Natural Selection Strategy. In: Kang, L., Liu, Y., Zeng, S. (eds.) ISICA 2007. LNCS, vol. 4683, pp. 334–343. Springer, Heidelberg (2007)
12. Deb, K.: Multi-Objective Optimization Using Evolutionary Algorithms. John Wiley and Sons (2001)
13. Esquivel, S.C., Coello, Coello, C.A.: On the use of particle swarm optimization with multimodal functions. In: Proc. Congr. Evol. Comput., pp. 1130–1136 (2003)
14. Shi, Y., Eberhart, R.C.: A modified particle swarm optimizer. In: Proc. IEEE World Congr. Comput. Intell., pp. 69–73 (1998)

Rank Based Hybrid Multimodal Fusion Using PSO

Amioy Kumar, Madasu Hanmandlu, Vaibhav Sharma, and H.M. Gupta

Biometrics Research Laboratory, Department of Electrical Engineering,
Indian Institute of Technology Delhi, Hauz Khas, New Delhi, India-110016

Abstract. This paper investigates a hybrid fusion methodology by incorporating *ranks* and *decisions* of a user in a multimodal biometric system using PSO. Each of the biometric modalities is first used to rank the users. The individual ranks are then integrated to get a fused rank for each user. The matching scores associated to the fused ranked users are employed to take accept or reject decisions for each modality. The final decision is made by integrating the two decisions by the individual modalities. The decision thresholds for two modalities and a decision level fusion rule are selected by incorporating PSO. The role of PSO is to adaptively choose the fusion parameters in the varying security needs by minimizing the error rates in the system. The proposed methodology has a particular importance when the scores associated with two modalities are in different domain and their integration on score level need extra complexity of normalization. The experimental results presented in this paper have shown the relevance of the proposed scheme.

Index Terms: Rank Level Fusion, Decision level fusion, Adaptive multimodal biometric management, and Particle Swarm Optimization.

1 Introduction

Biometric Security evolved as an ever increasing and promising technology for authentication due to its reliable and very accurate personal authentication. In recent decades, multimodal biometric systems have been investigated with better performance than unimodal systems [1]. A multimodal system can be categorized on the basis of level of fusion of the biometric modalities. In *feature level* fusion, features extracted from unimodal systems are first integrated [1] and the resultant features are then utilized in further authentication process. In the *score level* fusion, the matching scores are first generated using individual features. The scores are further integrated using fusion rules [1] and then used for authentication. The *decision level* is the highest level of fusion where the decisions (0 for 'reject', 1 for 'accept') taken using individual modalities are first fused using binary fusion rules [2] for final decision. When the output of each biometric matcher is a subset of possible matches sorted in the decreasing order of confidence, fusion can be done at the *rank level* [1].

The rank level fusion is relatively new and less researched problem in the biometric literature. One of the recent yet significant works on the rank level fusion is by Monwar et al in [4] and kumar et al in [5]. A summary of prior works on rank level

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 217–224, 2011.
© Springer-Verlag Berlin Heidelberg 2011

fusion is presented in [4]-[5]. The work in [4] opt for the Borda count, highest rank, and logistic regression methods for integrating the ear, face and voice biometrics while in [5] for multiple features in palmprint. However, most of the presented work operates on fixed threshold strategy and is not adaptive to the varying security requirements [2]. In such conditions, the need is to devise an optimization technique, which can optimally select fusion parameters according to the user defined security demand. The management of such adaptive multimodal systems is achieved using Particle Swarm Optimization (PSO) in [2][3]. In [2], the decision level fusion approach is proposed which is further improved by [3] using score level fusion. Even then, neither of the proposed work employed rank level fusion strategy in the adaptive management of the multimodal system.

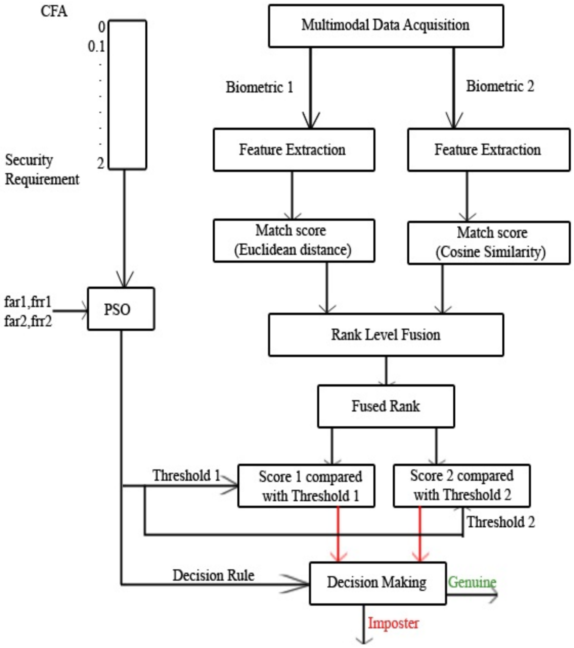


Fig. 1. Block diagram of the proposed system

We therefore, propose the usage of rank level fusion in the adaptive multimodal biometric fusion. The presented approach can be dividing into three steps: (1) each enrolled user is first ranked by each involved modalities in the multimodal system (2) the best ranked users are then used to take individual decisions by using respective matching scores and the decision thresholds. (3) The individual decisions are then fused using a binary fusion rule based on the binary combinations of the decision outcomes of multimodal system [2]. The optimal selection of individual decision thresholds and a binary fusion rule is done using PSO. The block diagram of the complete system is given in Fig. 1. The proposed system is fully automatic and can be applied for online security applications.

2 The Proposed Fusion Methodology

In the proposed method of multimodal fusion, the features extracted from biometric modalities are used to rank the enrolled users [4].The features extracted for each user is matched with others. The computed matching scores are arranged in the ascending order and the ranks are assigned based on the order of matching scores calculated for a user. Rank 1 is given to a user having the *best match*. The first part of the proposed algorithm is to integrate the ranks assigned by the individual modalities.

A. Rank Level Fusion

The goal of the rank-level fusion is to integrate the rank assigned by the individual biometric matchers in order to derive a consensus rank for each identity. The rank level fusion algorithm has already been utilized in earlier efforts [4]-[5]. Novel non-linear rank level fusion methods like: Tan hyperbolic (Tanh) and exponential sum is utilized in [5] as a potential rank level fusion algorithms. As high results demonstrated using Boarda count and highest rank method, we tested them for our experiments.

Let $r_{i,j}$ be the rank assigned to user j in the database by the i^{th} matcher, where i and j = 1,2…N; where N is the number of users. Let R_j is the final rank for user j after applying the rank level fusion. In the Borda count method, the fused rank is estimated as the sum of the ranks of individual classifiers:

$$R_j = \sum_{i=1}^{N} r_{i,j} \tag{1}$$

In the highest rank method, the consolidated rank is calculated using the minimum of all the associated ranks as :

$$R_j = \min(r_{i,j}) \text{ for all } i=1 \text{to} N \tag{2}$$

B. PSO Based Fusion Parameters

The next step of the proposed algorithm is to make the decisions from the fused ranks of the users. The matching scores and the corresponding error rates: False Acceptance Rate (FAR) and False Rejection Rate (FRR), corresponding to each modality are calculated as in [2]. PSO is then utilized to learn the decision thresholds by varying the cost of FAR (CFA) and FRR (CFR). The objective function of PSO is given as:

$$\begin{aligned} \text{Minimize} \quad & E = CFA * FAR + CFR * FRR. \\ & CFR = 2 - CFA \end{aligned} \tag{3}$$

In PSO each particle is associated with some velocity according to which it moves in the multi-dimensional solution space and it also has memory to keep information of its previous visited space; hence its movement is influenced by two factors: the local best solution due to itself and the global best solution due to all particles participating in the solution space. Let us consider a d-dimensional solution space. The i^{th} particle of the swarm can be represented as d-dimensional vector, $X_i = (x_{i1}, x_{i2}, .., x_{id})$ where the first subscript denotes the particle number and the second subscript denotes the dimension. The particle's motion is affected by its own best position denoted by Pos_{id} and the global best position denoted by Pos_{gd}. The velocity of a particle at k th iteration is updated by:

$$V_{id}^{k+1} = \omega V_{id}^{k} + r_1 \alpha \left(Pos_{id} - X_{id}^{k} \right) + r_2 \beta \left(Pos_{gd} - X_{id}^{k} \right) \tag{4}$$

The corresponding position of the particle is updated by:

$$X_{id}^{k+1} = X_{id}^{k} + V_{id}^{k+1} \tag{5}$$

where i = 1,2,3…..M; M being the number of swarms and d=1,2,3,…..N is the dimension of a swarm; α and β are the positive constants, called cognitive parameter and social parameter respectively.

In the proposed work, a set of values of FAR and FRR calculated for individual modalities, and the binary fusion rules [2] are used as input in the PSO algorithm. The PSO framework is same as in [2]-[3]. The output of the algorithm is the optimal FAR, FRR, and a binary fusion rule. Decision thresholds are calculated from the corresponding FAR and FRR which are finally stored to use at the final decision. PSO actually provides a framework to the multimodal system to make it adaptive to the varying security conditions incorporated by CFA and CFR which expected to be user defined in the real life applications.

C. The hybrid Fusion

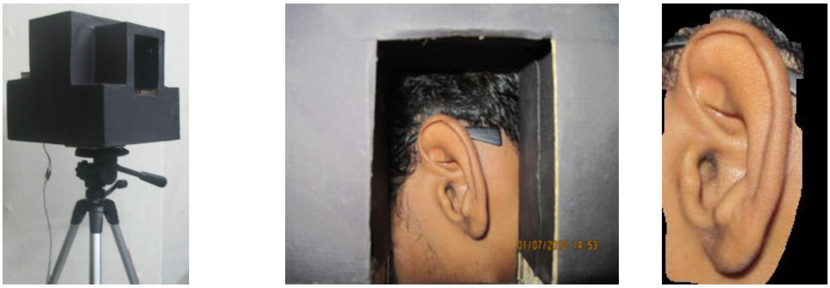
The final step in the proposed algorithm is to make a final decision by incorporating the learned decision thresholds for each modality and the binary fusion rule. Unlike the decision level fusion in [2] or the score level fusion in [3], here a hybrid of the rank and the decision level fusion is proposed for integration in the multimodal system. The matching scores corresponding to the fused ranked users are computed and the learned decision threshold is applied on the scores to get respective decisions. The final decision is taken by employing decision level fusion rules. The complete algorithm is given as:

```
Start
    I. Compute rank $R_j$ for each user j
    II. Generate fused rank $R_F$ using any rank fusion method
    III. Find the FAR and FRR for each modalities
    IV. Input the desired cost for security (CFA/CFR)
    V. Use PSO algorithm to learn the optimal decision threshold for each modalities and a binary fusion rule.
    VI. For Both Modalities
        ➢ Compute the minimum matching score (corresponding to the fused rank) by comparing to all the users.
        ➢ Match the score with the learned threshold for individual decisions.
        ➢ Get final decision by employing fusion rule on individual decisions
End
```

3 Experimental Results

The proposed rank based hybrid fusion methodology is tested on a multimodal database of ear and iris biometrics. The growing attention for using ear as a biometric modality is due to its rich and stable structure, and high user convenience. Ear database of 100 users with 5 samples each is created at biometrics research laboratory, IIT Delhi. The imaging setup and a sample image are shown in Fig. 2 (a) and 2 (b) respectively. The Gaussian classifiers are used to first segment the skin and non-skin areas in the ear images. Laplacian of Gaussian is then used to compute edges of the skin areas; which helped to get ear-ROI images. A localized region based active contour is finally located in the ear-ROI images [6]. The ear- contours are then employed for the authentication using SIFT features. The extracted ear image is shown in Fig. 2 (c). The SIFT based feature extraction is same as in [7]. The SIFT features extracted from a sample ear image is given in Fig. 3.

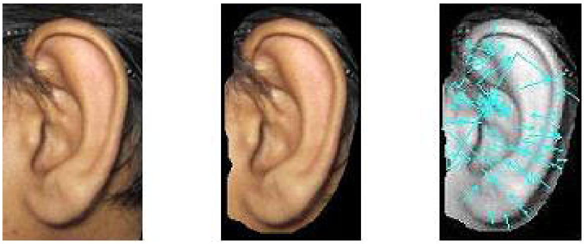
The CASIA V2 lamp database is used for iris images and the normalization of the iris strip is same as in [7]. A sample image and the corresponding normalized iris strip are given in Fig. 4. 100 users from iris database are paired with ear database to make multimodal system. The Euclidean distance is used to generate matching score for ear images while normalized Hamming distance for iris strips [8]. The rank associated by each modality and rank fusion using the Borda count and highest rank is same as in [4].The combined ROC curves for ear and iris databases, and their Borda count and highest rank based fusion is given in Fig. 5 (a). The FAR =0.05 and FRR=8% is reported for iris database. In comparison, ear has low performance as: FAR=1% and FRR=18%. The rank level fusion using Borda count executed on much better performance than individuals with FAR=0.06% and FRR=1.5%. However, the highest rank fusion scheme has inferior FAR=0.8% and FRR=2% (approax).



(a) (b) (c)

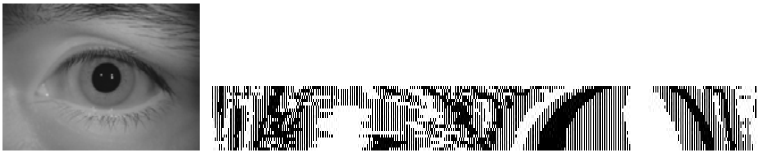
Fig. 2. Ear Acuisition: (a) Camera setup (b) Sample image (c) Extracted ear image

The Borda count based rank fusion is therefore carried out in the proposed hybrid fusion scheme. PSO is employed to learn the decision thresholds for each modality, and a binary fusion rule [2]. The matching scores of the best ranked users in the Borda count method are tested against respective decision thresholds.



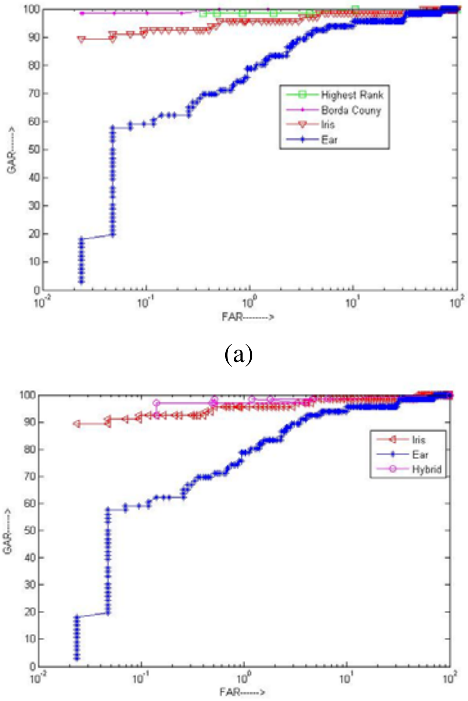
(a) (b) (c)

Fig. 3. (a) sample image (b) Normalized contour (c) SIFT features



(a) (b)

Fig. 4. (a) sample image (b) Normalized strip enhanced with Log Gabor filter

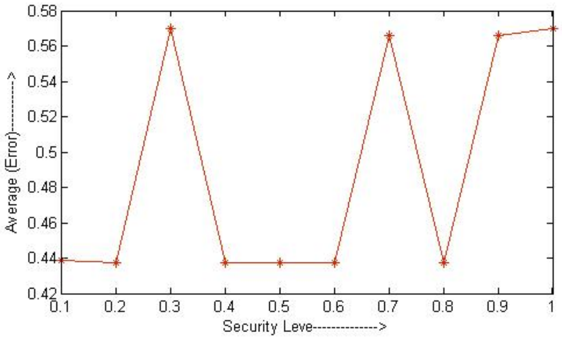


(a)

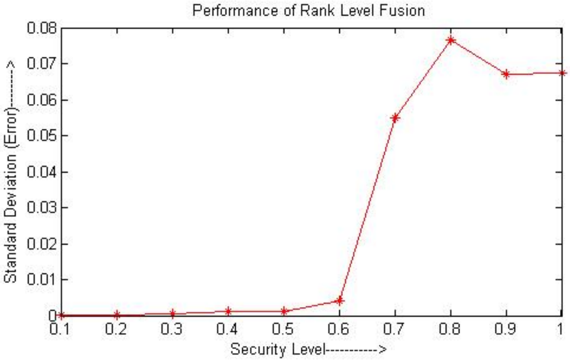
(b)

Fig. 5. ROC Curves: (a) Rank Level fusion schemes (b) Proposed hybrid fusion

The final decisions corresponding to each modality is fused using the learned binary fusion rule. The ROC curves for the proposed method with ear and iris modalities are given in Fig. 5 (b). The proposed fusion achieved FAR=0.09% and FRR=1.5% which is comparable to other rank level fusion techniques in [4]-[5]. The adaptive management of the fusion scheme is tested by incorporating varying security levels as in [3]-[4]. However, out of 16 binary rules for two sensors we found only AND rule is selected most of the times. Thus, in the final stages of the algorithm we fixed AND rule and the binary fusion rule and allowed the swarms to select only optimal decision thresholds corresponding each biometric modality with varying security levels. This strategy is also helpful in reducing the overall time complexity in learning fusion parameters from PSO. The average and standard deviations of the minimized errors in Eqn (3) is given in Fig. 6 (a) and 6 (b) respectively.



(a)



(b)

Fig. 6. Adaptive security analysis: (a) Average error (b) Standard deviation (error)

4 Conclusions

A novel rank based hybrid fusion methodology is investigated in this work. The involved modalities are used to rank each user which is integrated using Borda count rule. The matching scores of the best ranked users are then matched against the decision thresholds and the corresponding decisions are integrated using a binary fusion rule. The decision thresholds for each modality and the binary fusion rule are learned using PSO.

The proposed approach is tested on a multimodal system using ear and iris biometrics which operates on FAR=0.09% and FRR=1.5% and comparable to any rank level fusion approach in the literature. The adaptive management of the fusion parameters is done using PSO and shown a consistent performance of the presented scheme. The adaptive fusion strategies are first time investigated for the rank level fusion which is applicable for many real time applications. The developed model for adaptive rank level can be extended to any number of modalities.

References

[1] Ross, A., Nandakumar, K., Jain, A.K.: Handbook of Multibiometrics. Springer, New York (2006)

[2] Veeramachaneni, K., Osadciw, L.A., Varshney, P.K.: An Adaptive Multimodal Biometric Management Algorithm. IEEE Trans. on Systems, Man, and Cybernetics—Part C: Applications and Reviews 35(3) (August 2005)

[3] Kumar, A., Kanhangad, V., Zhang, D.: A New Framework for Adaptive Multimodal Biometrics Management. IEEE Trans. on Information Forensics and Security 5(1), 92–102 (2010)

[4] Monwar, M.M., Gavrilova, M.L.: Multimodal Biometric System Using Rank-Level Fusion Approach. IEEE Transaction on System, Man, and Cybernatics- Part B: Cybernatics 39, 867–878 (2009)

[5] Kumar, A., Shekhar, S.: Palmprint Recognition using Rank Level Fusion. In: Proceedings of 2010 IEEE 17th International Conference on Image Processing, Hong Kong, September 26-29, pp. 3121–3124 (2010)

[6] Lankton, S., Tannenbaum, A.: Localizing Region-Based Active Contours. IEEE Transactions on Image Processing 17(11), 2029–2039 (2008)

[7] Lowe, D.G.: Object recognition from local scale-invariant fea-tures. In: International Conference on Computer Vision, Corfu, Greece, pp. 1150–1157 (1999)

[8] Kumar, A., Passi, A.: Comparison and combination of iris matchers for reliable personal authentication. Pattern Recognition 23(3) (March 2010)

Grouping Genetic Algorithm for Data Clustering

Santhosh Peddi and Alok Singh

Department of Computer and Information Sciences,
University of Hyderabad, Hyderabad 500 046, India
peddi.santhosh@gmail.com
alokcs@uohyd.ernet.in

Abstract. Clustering can be visualized as a grouping problem as it consists of identifying finite set of groups in a dataset. Grouping genetic algorithms are specially designed to handle grouping problems. As the clustering criteria such as minimizing the with-in cluster distance is high-dimensional, non-linear and multi-modal, many standard algorithms available in the literature for clustering tend to converge to a locally optimal solution and/or have slow convergence. Even genetic guided clustering algorithms which are capable of identifying better quality solutions in general are also not totally immune to these shortcomings because of their ad hoc approach towards clustering invalidity and context insensitivity. To remove these shortcomings we have proposed a hybrid steady-state grouping genetic algorithm. Computational results show the effectiveness of our approach.

Keywords: Grouping Genetic Algorithm, Data Clustering, Heuristic.

1 Introduction

Clustering is the task of identifying finite set of groups in the dataset where similar instances are assigned to the same group and dissimilar ones to different groups. Clustering plays a vital role in data mining applications such as scientific data exploration, information retrieval and text mining. It also finds applications in spatial database applications, web analysis, customer relationship management (CRM), marketing, data compression, computational biology and many other related areas [1].

Clustering is among heavily studied problems and numerous algorithms have been proposed for it. Metaheuristic based approaches, in general, obtain better quality solutions in comparison to hill climbing based local search methods. This is because clustering criterions are normally high-dimensional, non- linear and multimodal [1, 2]). Number of genetic algorithms based approaches have already been proposed for clustering ([2], [3], [4], [5], [6], [7], [8], [9]). These approaches handle the problems of context insensitivity and clustering or grouping invalidity [10] in their own ad hoc ways. Removing the recombination operator from the genetic algorithm is a popular way of avoiding these problems [3]. However, removing the recombination operator can severely impair the search capability of genetic algorithms and can slow down their convergence drastically. To alleviate this problem, use of one-step K-means operator is suggested by Krishna and Murthy [3]. However, one-step K-means operator has its own problems [11]. Hong and Kwong [2] proposed a steady-state genetic

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 225–232, 2011.
© Springer-Verlag Berlin Heidelberg 2011

algorithm where traditional crossover operator is replaced with an ensemble learning operator based on average ling agglomerative clustering [12]. This algorithm was termed genetic-guided clustering algorithm with ensemble learning operator (GCEL). GCEL was compared with the approaches proposed in [3] and [4] and it out performed both these approaches.

Clearly clustering is a grouping problem [10], i.e., a problem where the aim is to group the instances of a dataset into one or more groups of instances, where each instance belong to exactly one group, in such a way that a given cost function is optimized. Grouping genetic algorithms are specially designed for grouping problems as traditional genetic algorithms suffer from context insensitivity and grouping invalidity while handling grouping problems [10]. So far no attempt has been made to use grouping genetic algorithm for data clustering. In this paper, we have proposed a steady-state grouping genetic algorithm for the data clustering. We have compared our approach against GCEL [2]. Computational results show the effectiveness of our approach.

The remainder of this paper is organized as follows: Section 2 describes our grouping genetic algorithm for data clustering. Computational results are presented in section 3, whereas section 4 outlines some concluding remarks and directions for future research.

2 The Grouping Genetic Algorithm

We have developed a hybrid steady-state grouping genetic algorithm (GGA) for the data clustering problem. Steady-state genetic algorithm uses steady-state population replacement method. Unlike generational replacement, where the entire parent population is replaced with an equal number of newly created children every generation, in the steady-state population replacement method a single child is produced in every generation and it replaces a less fit member of the population. In comparison to the generational method, the steady-state population replacement method generally finds better solutions faster. This is because of permanently keeping the best solutions in the population and the immediate availability of a child for selection and reproduction. Another advantage of the steady-state population replacement method is the ease with which duplicate copies of the same individuals are prevented in the population. In the generational approach, duplicate copies of the highly fit individuals may exist in the population. Within few generations, these highly fit individuals can dominate the whole population. When this happens, the crossover becomes totally ineffective and the mutation becomes the only possible way to improve solution quality. Under this situation improvement, if any, in solution quality is very slow. Such a situation is called the premature convergence. In the steady-state approach we can easily avoid this situation by simply checking each newly generated child against current population members and discarding it if it is identical to any member.

We have strings-of-group encoding strategy [2, 3] to represent a chromosome. In this encoding scheme, the value of an integer in the string is the label of the group to which the instance belongs. For example, consider a dataset containing 6 instances and if the number of clusters are fixed at two, then a chromosome (1 2 1 2 1 1) represents a solution where 1st, 3rd, 5th and 6th instances belong to one group and 2nd and 4th instances belong to another group. We have used the same fitness function as used in [2], so that we can directly compare our approach with the one proposed there. The objective of this fitness function is to minimize the within-cluster variation. Consider a clustering solution C = {C_1, C_2,...., C_K) of a data set D containing n unlabeled instances x_1, x_2,..., x_i,..., x_n each of which has m features, i.e., $x_i = (x_{i1}, x_{i2}, ..., x_{im})$ for i=1, 2,…, n, then the within-cluster variation *f*(C) of C is calculated as follows:

$$f(C) = \sum_{i=1}^{n} \sum_{j=1}^{K} \delta(x_i, C_j) \sum_{k=1}^{m} (x_{ik} - c_{jk})^2$$

where

$$\delta(x_i, C_j) = \begin{cases} 1 \text{ if the instance } x_i \in C_j \\ 0 \text{ otherwise} \end{cases}$$

and

$$c_{jk} = \frac{\sum_{i=1}^{n} \delta(x_i, C_j).x_{ik}}{\sum_{i=1}^{n} \delta(x_i, C_j)}$$

Here, we have followed the notational conventions of [2].

We have used a crossover operator that is derived from the crossover operator proposed in Singh and Gupta [13] for the one dimensional bin packing problem. This crossover operator does not suffer from the problem of clustering invalidity and context insensitivity as it operates on the clusters instead of individual instances. Let A and B be two parents selected for crossover and let each solution has M clusters. The crossover operator, we have used is described by following pseudo code, where new_sol is the child solution being generated by crossover and μ_{01} is a uniform variate.

```
i ← 1
new_sol ← ∅
while (i ≤ M) {
        if ($\mu_{01}$ < 0.5)
                select the best cluster C from A
        else
                select the best cluster C from B
        remove the instances belonging to C from both A and B
        new_sol ← new_sol U {C}
        i ← i + 1
}
```

In order to identify the best cluster in a parent, the fitness value of a cluster is divided by the number of instances assigned to that cluster. If this is not done then a cluster containing few instances will be preferred by the crossover over a bigger good cluster. This could have affected the performance of crossover operator described above adversely. While performing the crossover operation, it might happen that some clusters become empty or may contain only a single instance. In such cases we have assigned an arbitrarily large value to that cluster as its fitness value. This is done to prevent the selection of such clusters in future iterations of our crossover operator.

After the application of crossover operator some instances may not be assigned to any cluster. Such unassigned instances are reassigned to clusters by a heuristic. The heuristic considers the unassigned instances one-by-one in their natural order and assigns the instance in consideration to a cluster in a way that leads to least increment in the value of the fitness function.

The mutation operator tries to make a copy of the selected chromosome. However, each instance of the selected chromosome is copied to the corresponding cluster of the child with probability p_{copy}, otherwise it is included into the list of unassigned instances. All these unassigned instances are assigned to clusters by following the same heuristic as used in the crossover operator.

The crossover operator and the mutation operator are applied in a mutual exclusive way during each iteration. The crossover operator is applied with a probability of p_c, otherwise mutation is used.

We have used the probabilistic binary tournament selection for selecting two parents for crossover and a single parent for mutation. In probabilistic binary tournament selection, two different chromosomes are selected uniformly at random from current population and the chromosome with a better fitness value is selected to be a parent with probability p_b, otherwise the worse among the two is selected as parent.

We have generated the initial population randomly. However, as our genetic algorithm is steady-state, we have therefore guaranteed the uniqueness of each member of the initial population. Each member of the initial population is generated in the same manner as in [2], i.e., each member is generated using the random subspace method [14] followed by K-means clustering algorithm.

Moreover, during each generation the newly produced child is compared with current population members, and it is discarded if it is either identical to any member or have fitness worse than the worst member of the population, otherwise, it is included in the population in place of the worst member.

3 Computational Results

We have used one synthetic dataset and five real datasets to test the performance of our proposed grouping genetic algorithm (GGA). For comparing our approach, we have also implemented genetic-guided clustering algorithm with ensemble learning operator (GCEL) [2], which is the best approach available in the literature. In all our computational experiments with grouping genetic algorithm we have used $p_c = 0.8$, $p_{copy} = 0.8$, $p_b = 0.9$. We have used a population size of 100 for all datasets except the synthetic dataset for which the population size is taken to be 20 due to the lack of sufficient diverse solutions. All parameter values for GCEL are same as used in [2]. To allow a fair comparison, the number of iterations was fixed to 10000 for both GGA and GCEL. These two algorithms were executed 10 independent times on each dataset. We report the average results of these 10 runs.

We have used one synthetic dataset viz. X8K5D, and five real datasets viz. Ionosphere, Promoters, Colon, Breast Tissue and Glass Identification to test the performance of GGA and GCEL. Most of these real datasets are high-dimensional datasets

Table 1. Datasets and their characteristics

Data set	**Instances**	**Nominal features**	**Numeric features**	**Classes**
X8K5D	100	0	8	5
Ionosphere	351	0	34	2
Promoters	106	57	0	2
Colon	100	0	17	7
Breast Tissue	106	0	10	6
Glass Identification	214	0	10	7

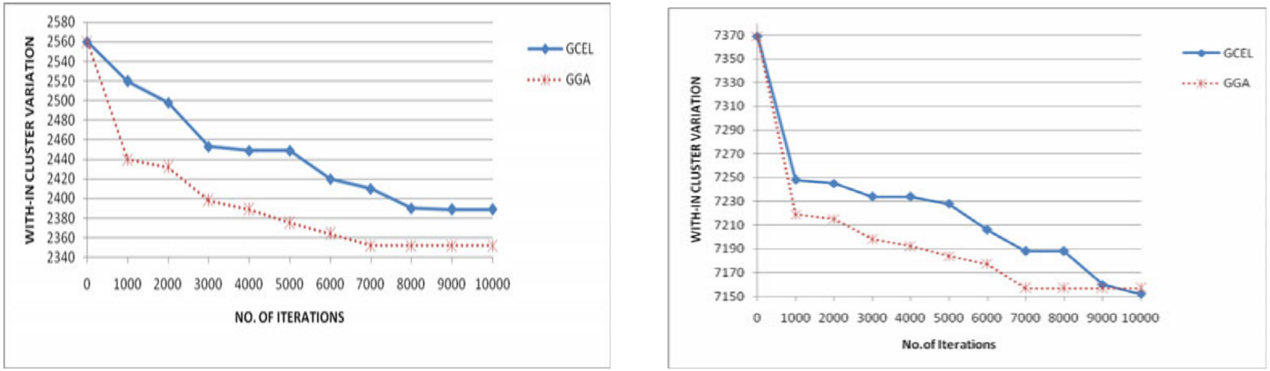
with small number of instances. The X8K5D dataset originally contains 1000 instances. However, like [2], we randomly chose the 100 instances from this dataset for use in our experiments. Table 1 presents the details of various data sets.

The average with-in cluster variations obtained by GGA and GCEL for all datasets are reported in Table 2. Also reported is the percentage improvement in solution quality of GCEL by GGA. Both GGA and GCEL obtained the same with-in cluster variation in all runs on X8K5D dataset. Except for promoter data set, where GCEL obtained better average with-in cluster variation, on all other real data sets GGA got better results. Percentage improvement of GGA over GCEL is highest on Glass Identification data set followed by Breast Tissue data set. Percentage improvement is negative for promoter dataset because GCEL obtained a slightly better result than GGA.

Figures 1(a), 1(b), 1(c) and 1(d) show the decrease in with-in cluster variations with number of iterations for GCEL and GGA on different data sets. These graphs clearly show that GGA converges to better quality solutions quickly in comparison to GCEL. Therefore, GGA is superior to GCEL not only in terms of solution quality but also in terms of rate of convergence.

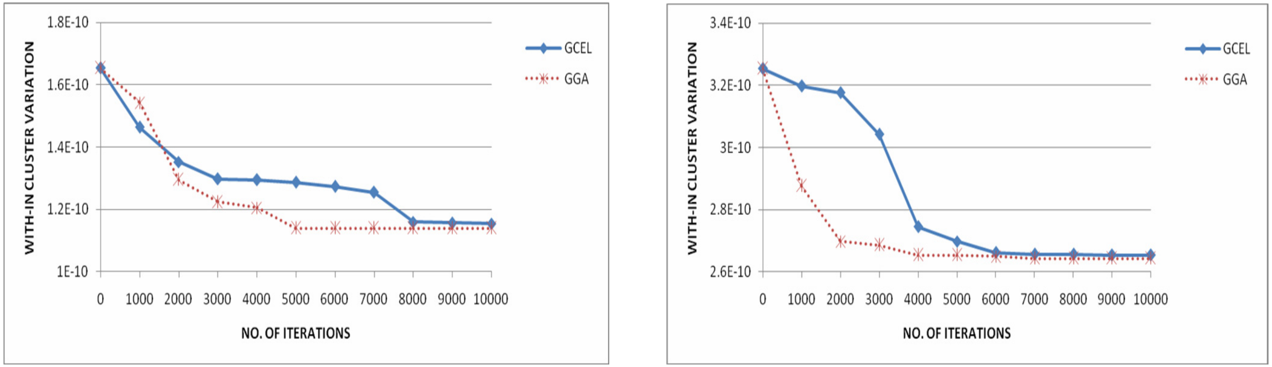
Table 2. Within-cluster variation

Data set	**GCEL**	**GGA**	**%Imp**
X8K5D	7.1496±0.000E+00	7.1496±0.000E+00	0.00%
Ionosphere	2.416±0.002E+03	2.352±0.426E+03	2.65%
Promoters	7.152±0.001E+03	7.176±0.023E+03	-0.33%
Colon	1.154±0.642E+10	1.139±0.472E+10	1.30%
Breast Tissue	3.254±0.142E+10	2.642±0.006E+10	18.81%
Glass Identification	4.964±0.263E+02	4.623±0.063E+02	6.87%



(a) Ionosphere Data Set

(b) Promoter Data Set



(c) Colon Data Set

(d) Breast Tissue Data Set

Fig. 1. Reduction in with-in cluster variation with iterations for GCEL and GGA on different datasets

4 Conclusions and Future Work

In this paper, we have proposed a grouping genetic algorithm for data clustering problem. Grouping genetic algorithm is specially designed for handling grouping problems. As data clustering problem is a grouping problem and since no attempt is made so far to solve the data clustering problem using grouping genetic algorithm, we made an attempt to solve the data clustering problem using grouping genetic algorithm. Experimental results on standard datasets showed that our algorithm gives better results in comparison to GCEL [2]. Our approach performed equally well on both synthetic and real datasets.

In the future, we would like to improve our approach by investigating in the following directions:

- By using different encoding schemes.
- By using different variations of crossover and mutation operators.
- By using different local search in our approach.

The basic ideas presented here can be utilized for developing other population based meta-heuristic approaches such as artificial bee colony algorithm, differential evaluation etc. for data clustering.

References

1. Jain, A.K., Murthy, M.N., Flynn, P.J.: Data clustering: a review. ACM Computing Surveys 13, 264–323 (1999)
2. Hong, Y., Kwong, S.: To combine steady state genetic algorithm and ensemble learning for data clustering. Pattern Recognition Letters 29, 1416–1423 (2008)
3. Krishna, K., Murthy, M.N.: Genetic K-means Algorithm. IEEE Transactions on Systems, Man Cybernetic - Part B 29, 433–439 (1999)
4. Painho, M., Fernando, B.: Using genetic algorithms in clustering problems. In: Proceedings of the 5th International Conference on GeoComputation (2000)
5. Kuncheva, L.I., Bezdeck, J.C.: Nearest prototype classification: clustering genetic algorithm or random search? IEEE Transactions on. Systems, Man and Cybernetics - Part B 28, 160–164 (1998)
6. Franti, P.: Genetic algorithm with deterministic crossover for vector quantization. Pattern recognition Letters 21, 61–68 (2000)
7. Garai, G., Chaudhury, B.B.: A novel genetic algorithm for automatic clustering. Pattern Recognition Letters 25, 173–187 (2004)
8. Mitra, S.: An evolutionary rough partitive clustering. Pattern Recognition Letters 25, 1439–1449 (2004)
9. Martnez-Otzeta, J.M., Sierra, B., Lazkano, E., Astigarraga, A.: Classifier hierarchy learning by means of genetic algorithms. Pattern Recognition Letters 27, 1998–2004 (2006)
10. Falkenauer, E.: Genetic algorithms and grouping problems. John Wiley & Sons, Chicester (1998)

11. Sheng, W., Tucker, A., Liu, X.: Clustering with Niching Genetic K-means Algorithm. In: Deb, K., et al. (eds.) GECCO 2004. LNCS, vol. 3103, pp. 162–173. Springer, Heidelberg (2004)
12. Fred, A., Jain, A.K.: Combining multiple clusterings using evidence accumulation. IEEE Transactions on Pattern Analysis and Machine Intelligence 27, 835–850 (2005)
13. Singh, A., Gupta, A.K.: Two heuristics for the one-dimensional bin-packing problem. OR Spectrum 29, 765–781 (2007)
14. Ho, T.K.: The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence 20, 832–844 (1998)

Genetic Algorithm for Optimizing Neural Network Based Software Cost Estimation

Tirimula Rao Benala[1], Satchidananda Dehuri[2],
Suresh Chandra Satapathy[1], and Ch. Sudha Raghavi[1]

[1] Anil Neerukonda Institute of Technology and Sciences
Sangivalasa, Visakhapatnam, Andhra Pradesh, India
b.tirimula@gmail.com, sureshsatapathy@ieee.org,
sudharaghavich@yahoo.com
[2] Department of Information & Communication Technology
Fakir Mohan University, Vyasa Vihar, Balasore-756019, India
satchi.lapa@gmail.com

Abstract. Software engineering cost models and estimation techniques are used for number of purposes. These include budgeting, tradeoff and risk analysis, project planning and control, software improvement and investment analysis. The proposed work uses neural network based estimation, which is essentially a machine learning approach, is one of the most popular techniques. In this paper the author has proposed a 2 step process for software effort prediction. In first phase known as training phase neural network selects the matching class (datasets) for the given input, which is improved by optimizing the parameters of each individual dataset by Genetic algorithm. In second step known as testing phase, the prediction process is done by adaptive neural networks. The proposed method uses COCOMO-II as base model. The experimental results show that our method could significantly improve prediction accuracy of conventional Artificial Neural Networks (ANN) and has potential to become an effective method for software cost estimation.

Keywords: Software cost estimation, Genetic algorithm, ANN, BP-Learning, and COCOMO-II.

1 Introduction

Software cost estimation is critical for the success of software project management. It affects management activities including resource allocation, project bidding and project planning. The importance of accurate estimation has led to extensive research efforts to software cost estimation methods. From a comprehensive review, these methods could be classified into following six categories: parametric models including COCOMO, SLIM and SEER-SEM, expert judgment including Delphi technique and work break down structure based methods, learning oriented techniques including machine learning methods and analogy based estimation; regression based methods including ordinary least square regression and robust regression; dynamics based model, composite methods. [5]

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 233–239, 2011.
© Springer-Verlag Berlin Heidelberg 2011

In this paper, we are concerned with cost estimation models that are based on artificial neural networks .The artificial neural network (ANN), inspired by biological nervous system, are nonlinear information processing models, which are built from interconnected elementary processing elements called neurons. In general for software cost estimation modeling, the most commonly adopted architecture, learning algorithm and the activation functions are respectively feed forward, multi layer perceptron and the Back propagation algorithm and the Sigmoid function. Many researchers have applied the neural networks approach to estimate software development effort. However some difficulties are still confronted by Neural Network techniques such as neural networks are better suited for classification and categorization problems whereas software cost estimation is more of generalization rather than classification problem [1, 4]; non-normal characteristics of software engineering data set. The large and non-normal data sets always lead ANN methods to low prediction accuracy and high computational complexity. To alleviate these drawbacks our proposed idea has been devoted to simultaneously optimize selected class of projects and their feature selection by genetic algorithm (GA).

In this paper we have incorporated Nasser Tadayon [9]'s adaptive learning based machine using neural networks to estimate the software cost using neural network. GA is used to optimize the selected class of projects and their feature weights. Neural network based techniques and cost estimation fundamentals are briefly reviewed in section 2. The proposed GA approach for optimizing the selected class of projects is described in section 3.In section 4, numerical examples from COCOMO dataset is used to illustrate the performance. A conclusion and overview of future work conclude this paper.

2 Background

2.1 The COCOMO

The Constructive Cost Model, COCOMO, was introduced by Boehm [3].It has become one of the most widely used software cost estimation models in the industry. To support new life cycles and capability, it has evolved into a more comprehensive estimation model, called COCOMO II [2]. COCOMO II consists of three sub models, each one offering increased fidelity. Listed in increased fidelity, these sub models are called Application Composition, Early design and Post Architecture models. Until recently, only the last and most detailed sub model, Post Architecture had been implemented in a calibrated software tool. As such, unless otherwise explicitly specified, all further references in this study to COCOMO II can be assumed to be the Post Architecture Model [7].

The cost effort equation was calibrated and developed by Barry Boehm using the COCOMO II [2] post architecture model for software cost estimation. It computes the efforts (in Person-Months) required for a project based on project's size in KSLOC (Kilo Source Lines Of Code) as well as 22 cost factors (5 Scale factors and 17 Effort Multipliers) by:

$$PM = A.(SIZE)^{1.01+\sum_{i=1}^{5} SF_i} . \prod_{i=1}^{17} EM_i \tag{1}$$

where A is a multiplicative constant, and the set of SF (Scale Factor), and EM (Effort Multiplier). There are 17 Effort Multipliers capturing the characteristics of the software development that affect the effort to complete the project. These weighted multipliers are grouped into four categories (Product, platform, personnel, and project), and their product is used to adjust the effort. The nominal weight assigned to each multiplier is 1.0. If a rating level has detrimental effect on effort, its corresponding multiplier is greater than 1.0. Conversely if the rating level reduces the effort then the corresponding multiplier is less than 1.0.There are five exponent scale factors (Precedenteness , development flexibility, architecture/risk resolution, team cohesion, and process maturity). They account for the relative economies or diseconomies of scale encountered as a software project increases its size based on based on different nominal values and rating schemes [7].

2.2 Neural Network Based Cost Estimation

The neural network architecture for software cost estimation is given as

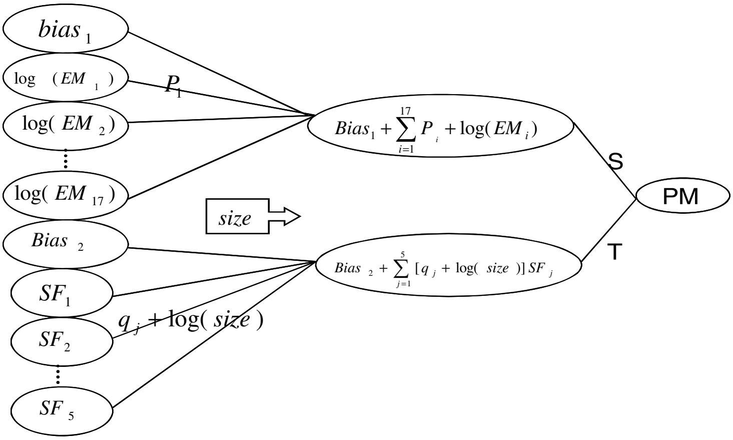


Fig. 1. Network Architecture

We compute the effort (PM) using the mathematical approach given by Tadayon [11].

3 Framework

Genetic algorithm (GA) is a stochastic global optimization technique initially introduced by Holland in 1970's [6]. By mimicking biological selection and reproduction, GA can efficiently search through the solution space of complex problems and it is naturally parallel and provides opportunity to escape from local optimum. GA has become one of the most popular algorithms for optimization problems. In this section, we construct the OCFWANN system (stands for Optimal projects of predicted Class and Feature Weighting and Artificial Neural Network based Estimation) which can perform simultaneous optimization of 'N' projects of the predicted class and their feature weights. GA is selected as optimization tool for OCFWANN system. The detailed description is presented in section 3.2. In order to introduce the fitness function in GA algorithm, performance metrics for estimation accuracy are firstly presented in the section 3.1.

3.1 Performance Evaluation Metrics

To measure the accuracies of the proposed methods, three performance metrics are considered: Mean Magnitude of Relative Error (MMRE), Median Magnitude of relative error (MdMRE),and PRED (0.25), because these measures are widely accepted in literature [6].

The MMRE is defined as:

$$MMRE = \frac{1}{n} \sum_{i=1}^{n} MRE \tag{2}$$

$$MRE = \left| \frac{(E_i - \hat{E}_i)}{E_i} \right| \tag{3}$$

Where n denotes the total number of projects, E_i denotes the actual cost of ith project, and $\hat{E}_i$ denotes the estimated cost of the ith project. Small MMRE value indicates the low level of estimation error. However this metric is unbalanced and penalizes overestimation more than underestimation. The MdMRE is the median of all the MREs.

$$\text{MdMRE} = \text{Median (MRE)} \tag{4}$$

It exhibits similar pattern to MMRE but it is more likely to select the true model especially in the underestimation case since it is less sensitive to extreme outlier [6]. The PRED (0.25) is the percentage of prediction that fall within 25 percent of actual cost

$$PRED(q) = \frac{k}{n} \tag{5}$$

Where n denotes the total number of projects and k denotes the number of projects whose MRE is less than or equal to q. Normally, q is set to be 0.25. The PRED(0.25) identifies cost estimations that are generally accurate, while MMRE is biased and not always reliable as a performance metric. However, MMRE has been de facto standard in the software cost estimation literature. Thus MMRE is selected for the fitness function in GA. More specifically, for each combination of ‘N’ project parameters and cost driver weights, MMRE is computed across the validation dataset. Then GA searches through the project parameter space to minimize MMRE.

3.2 GA for Optimization

The procedure for OPFWANN system via Genetic Algorithm is presented in this section. The system consists of two stages: the first one is training stage and the second one is testing stage. In the training stage 93 NASA data points are presented to the system, the ANN is configured with cost drivers to produce the cost prediction for the given input data point. The class label of the input data is determined basing on PM obtained. GA explores the class space to minimize the error (in terms of MMRE) of ANN on the training projects by the following steps:

3.2.1 Encoding

To apply GA for searching, the cost drivers are encoded as binary string chromosome. Each individual chromosome consists of number of binary digits. There are six features for each driver (very low- vl, low-l, nominal-n, high-h, very high-vh, extra high-xh). Here we encode cost driver weights with 3 bits (0-n,1-vl,2-l,3-n,4-h,5-vh,6-xh,7-n). 0 and 7 are assumed default values nominal(n). Since the cost driver weights are decimal values, before entering into ANN model these binary codes are transformed into decimal numbers.

3.2.1.1 Population Generation and fitness function. After encoding the individual chromosome, the system then generates a population of chromosomes. Each chromosome is evaluated by the fitness function in GA. Since the GA is designed to maximize the fitness value and the smaller MMRE value indicates more accurate prediction, we set the fitness function as the reciprocal of MMRE.

$$f = \frac{1}{MMRE} \tag{6}$$

Given one training project as input, ANN predicts the PM for the project, basing on the person month, the class is identified for the project, which contains set of similar projects as input. To evaluate the prediction performance of the ANN model, the error metric MMRE, PRED (0.25), and MdMRE applied on the training project set in the class. Then, the reciprocal of MMRE is used as the fitness value for each cost driver combination (or chromosome).

3.2.2 Rules for Selection, Extinction and Multiplication

The standard roulette wheel is used to select chromosomes from the current population. The selected chromosome were consecutively paired with a probability of 0.8 was used to produce new chromosome in each pair. The newly created chromosome constituted a new population. The population is evolved by GA algorithm using evolutionary rules described above. The individual with best fitness value is in the population in every cycle.

3.2.3 Completion of Evaluation

The population is evolved by the GA algorithm in the first stage until the number of generations is equal to or excess 2000 or the best fitness value did not change in the last 200 generations. The second stage is the testing stage. In this stage the system receives the optimized parameters from the training stage to configure the ANN model. The optimal ANN is then applied to the testing project to evaluate the performance of the trained ANN.

4 Experimentations and Results

The COCOMO NASA 2 Dataset containing 93 data points have been taken for our experiment [8]. The data is in COCOMO-I format calibrated to COCOMO-II using Rosetta stone [2]. COCOMO measures efforts in the calendar months of 152 hours (and includes development and management hours). COCOMO assumes that the effort grows more than linearly on software size. There are total of 17 effort multipliers in COCOMO II. These cost factors are expressed in 6 forms i.e vl ,l ,n, h, vh ,hx. For our experimental results we have included 5 scaling factors and assumed their values as "nominal". Along with these there are two more attributes, namely, KSLOC (Kilo Source Lines of Code) and actual effort. There are 11 classes distributed across 93 data points.

For the purpose of validation, we adopt three-fold cross validation [6] to evaluate accuracy of the methods. In this scheme the NASA dataset is randomly divided into three nearly equal sized subsets. At each time one of three subsets is used as the test sets which is exclusively used to evaluate the estimation performance, and other two subsets treated as Validation data set and training data set exclusively used to optimize the cost drivers. This process is repeated three times. Then the average training error and testing error across all three trials are computed. The advantage of this scheme is it matters less how the data is split since each data point is assigned into a test set, a training set and a validation set respectively once. In the experiment we apply three types of ANN based models. The first model is conventional ANN [11], the second model is FWANN (GA optimizing feature weights(cost drivers) for Neural Network based cost estimation) which does not optimize the projects data points in the corresponding class. The third model is OCFWANN which uses GA simultaneously optimize the class and the feature weights (cost drivers).For comparison, other popular estimation models including Analogy Based Estimation(ABE), Step wise regression(SWR)[9],Artificial Neural Networks(ANN)[11],classification and regression trees (CART) [10], are also included in the experiments.

The experimental results are summarized in table 2. It shows that OCFWANN achieves the best level of prediction performance (0.32 for MMRE, 0.24 for MdMRE, and 0.31 for PRED (0.25)).

Table 1. Results and Comparisons on NASA Dataset

Methods	MMRE		MdMRE		PRED(0.25)	
	Training	Testing	Training	Testing	Training	Testing
OCFWANN	0.33	0.32	0.29	0.24	0.36	0.31
OFWANN	0.43	0.38	0.44	0.33	0.35	0.43
ANN	0.48	0.42	0.42	0.38	0.51	0.44
SWR	0.92	0.79	0.58	0.44	0.45	0.41
CART	0.85	0.64	0.48	0.37	0.34	0.30

5 Conclusion and Future work

On appraising the above novel technique the hybrid system of GA and ANN provides better prediction accuracy compared to ANN. GA is used as a tool for simultaneously optimizing the concerned class to which the input project belongs and cost drivers. The experimental results show that our method gives pacifying optimal performance as compared to conventional ANN and outperform the comparative techniques such as OFWANN, SWR and CART. Motivation is therefore exploring the scope of application of soft computing in the field of Software Cost Estimation.

We have done our research in the direction of software cost estimation by hybrid system using GA as it has not been explored extensively till date. There are numerous cost estimation techniques that have been proposed in different real-world applications. We extend connotations to our work with Artificial Bee Colony (ABC), Differential Evolution (DE), Artificial Immune System (AIS), Bacterial foraging optimization algorithm, Neuro Fuzzy, Neuro Genetic, Simulated Annealing and fuzzy logic.

References

1. AttarZadeh, I., Ow, S.H.: Proposing a New Software Cost Estimation Model Based on Artificial Neural Networks. In: IEEE 2nd Conference on Computer Engineering and Technology, ICCET (2010) ISBN 978-1-4244-6349-7/10
2. Boehm, B., Abts, C., Brown, A., Chulani, S., Clark, B., Horowitz, E., Madach, R., Reifer, D., Steece, B.: Software Cost Estimation with COCOMO II. Prentice Hall, Upper Saddle River (2000)
3. Boehm, B.: Software Engineering Economics. Prentice-Hall (1981)
4. Idri, A., Khoshgoftaar, T.M., Abran, A.: Can Neural Networks be easily interpreted in Software Cost Estimation? In: FUZZ-IEEE 2002, Proceeding of IEEE International Conference on Fuzzy Systems (2002) ISBN 0-7803-7280-8/02
5. Li, Y.F., Xie, M., Goh, T.N.: A study Of Project Selection Feature Weighting For Analogy Based Software Cost Estimation. The Journal Of Systems and Software 82, 241–252 (2009)
6. Li, Y.F., Xie, M., Goh, T.N.: Optimization of Feature Weights and Number of Neighbors For Analogy Based Cost Estimation in Software Project Management. In: Proceedings of the 2008 IEEE IEEM (2008) ISBN: 978-1-4244-2630-0/08
7. Musilek, P., Pedrycz, W., Sun, N.: On the Sensitivity of COCOMO II Software Cost Estimation Model. In: METRICS 2002, The Proceedings of 8th IEEE Symposium on Software Metrics (2002) ISBN 0-7695-1339-5/02
8. Menzies, T.: The PROMISE Repository Of Software Engineering Databases, School Of Information Technology and Engineering, University of Ottawa, Canada (2006), `http://promise.site.uottawa.ca/SERepository`
9. Shepperd, M., Kadoda, G.: Comparing Software Prediction Techniques using Simulation. IEEE Transaction on Software Engineering 27(11), 1014–1022 (2001)
10. Stensrud, E.: Alternative Approaches to Software Prediction of ERP Projects. Information and Software Technology 43(7), 413–423 (2001)
11. Tadayon, N.: Neural Network approach for Software Cost Estimation. In: ITCC 2005, Proceeding of IEEE International Conference on Information Technology: Coding and Computing (2005) ISBN 0-7695-2315-3/05

IAMGA: Intimate-Based Assortative Mating Genetic Algorithm

Fatemeh Ramezani[1] and Shahriar Lotfi[2]

[1] Computer Enginering Department, College of Nabi Akram, Iran
[2] Computer Science Department, University of Tabriz, Iran
f60_ramezani@yahoo.com, shahriar_lotfi@tabrizu.ac.ir

Abstract. Standard Genetic Algorithms (SGAs) is modeled as a simple set of fixed size individuals and each individual has no gender. The idea is based on non-random mating and important role of religious in the organization of societies. Essential concepts of religions are commandments and religious customs, which influence the behavior of the individuals. This paper proposes the Intimate-Based Assortative Mating Genetic Algorithm (IAMGA) and explores the affect of including intimate-based assortative mating to improve the performance of genetic algorithms. The IAMGA combined gender-based, variable-size and intimate-based assortative mating feature. All mentioned benchmark instances were clearly better than the performance of a SGA.

Keywords: Standard Genetic Algorithm, Gender-Based Genetic Algorithm, Gender, Assortative Mating, Intimate Relationship and Infusion Operator.

1 Introduction

Genetic Algorithms are the inspiration in the natural evolution. Over many generations, natural populations evolve according to the principles of “natural selection” and “survival of the fittest”. These principles were clearly stated by Charles Darwin in the Origin of the Species [1] and [2].

Genetic algorithms encode possible solutions for a problem as a “population” of simple chromosome-like data structures and apply recombination operators to these structures to generate “descendants” that are joined in new populations. If the solutions are properly encoded, each new generation contains “better adapted” chromosomes, optimizing the solution [3]. John Holland [2] established the basic principles of genetic algorithms.

Although GAs were introduced to modeling elements of natural evolution, one of the most crucial key, was left out. It was named gender. Through gender came the developing of sexual selection, a component of natural selection where reproductive success depends on interaction with a partner of the opposite sex to produce offspring [4]. The idea to exploit genders in genetic algorithms has been considered before in various publications, inspired by nature’s example [5].

Marriage is a concept to generate new generation in human religious communities; humans wed with different sex to generate new family. In these societies marriage with intimates is forbidden. This paper surveys new hypothesis in GAs. It describes a

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 240–247, 2011.
© Springer-Verlag Berlin Heidelberg 2011

new relationship among genders called "Intimate Relationship" and defines new crossover mechanism for GAs. This mechanism is based on the religious marriage in human society that is the choice of a mate based on certain characteristics. To apply this mechanism, the population is divided into female and male genders.

The paper is organized as follow: Section 2 highlights IAMGA differences with respect to the classical Holland's algorithm. In Section 3, some related work is presented. In Section 4, proposed new approach and the motivation of the IAMGA algorithm is presented. The obtained results and the comparison with the classical algorithm are shown in Section 5 and numerical experiments and results are presented.

2 Background

The GA algorithms are stochastic methods for searching and finding best solutions [6]. This section begins with describing Standard Genetic Algorithms (SGAs) and continues with some background on biological and social keys of human society such as intimate relationship and gender.

2.1 Standard Genetic Algorithms (SGAs)

John Holland first introduced SGAs for the formal investigation of the mechanisms of natural adaptation [2].Algorithm starts with a set of chromosomes called population. Solutions are taken and used to form a new population. Solutions are selected according to their fitness; the more suitable they are the more chances they have to reproduce, to form new offspring. It consists of three main operators: reproduction, crossover and mutation [7].

2.2 Intimate Relationship

In different divine religions people are forbidden to marry whose have an intimate relationship. Marriage with ancestor and child of ancestor is banned; such as uncles, nephews or nieces.

2.3 Gender

Male and female are different gender group in nature. Human cells contain 23 pairs of chromosomes for a total of 46. There is one pair of sex chromosome and the other pairs are autosomes. The Gender separation and sexual reproduction have been interest in many studies and application of genetic algorithm (GAs), since they are an important feature of the living organisms [8].

3 Related Works

Mate choice idea is not new and had been incorporated in SGA. Studies on Gender-based GAs can be found in [8-11] but usually the inclusion of gender is merely limited in multi-objective optimization or as a tag in the chromosome-preventing crossover with other individuals bearing the same gender flag.

Similarly, Ratford [12-13] and Ronald [14] both proposed selection schemes in which the first mate is selected using a traditional selection method, with the second mate being selected based on some seduction function between itself and the first mate.

Rejeb and Abuelhaija [10] proposed a method that adds the gender feature to chromosomes by representing “1” for male and “0” for female. They use a gendered genetic algorithm to solve graph-partitioning problems.

Song Goh, Lim and Rodrigues [15] proposed the new Sexual Selection Scheme. They suggest that mate choice in some species operates through female choice. They are determining the sex of individuals randomly or based on some problem-specific knowledge and involve the actual selection of a pair of individuals (one male and one female) hence the selection scheme would become problem dependant.

Parent-centric real-parameter crossover operators, proposed by Martinez and Lozano in [16], create the offspring in the neighborhood of the female parent by using a probability distribution and the male one defines the range of this probability distribution. The female and male differentiation process determines the individuals become female or/and male parents.

Vrajitoru [8] proposed four types of individuals: male (M), female (F), self-fertilizing (S-F), and hermaphrodite (H).

Ansotegui, Sellmann and Tierney [5] proposed to apply different selection pressure on the two gender populations. They apply intra-specific competition only in one part of the population and use Gender-based Genetic Algorithm (GGA) for the automatic configuration of solvers.

Wagner and Affenzeller [17] introduced a new sexual selection for Genetic Algorithms based on the concepts of male vigor and female choice of population genetics which provides the possibility to use two different selection schemes simultaneously within one algorithm.

4 Intimate-Based Assortative Mating Genetic Algorithm

In normal single/multiple-point crossover technique each mating of a couple of individuals creates a couple of offspring but in nature, in common, just mating of a couple of male and female individuals is allowed. Moreover in human religious community mating of species with intimate relationship is banned.

Although SGAs were introduced to modeling elements of natural evolution but there is no implicit notion of separate sexes but gender separation in SGAs is not a new idea. The main objective of this study is to show the performance of the gender based individual and intimate based mating.

The population of solutions consists of gender separation: male and female individuals. In the initialization, generate male and female individuals with equal probability 50% but over generations the numbers of male and female individuals are not equal.

In selection operation, mating couples are formed by selecting the individuals from the female/male population like in SGA. In crossover operation, the parents are chosen from the intermediate population by using the male selection. Crossover probability is used to measure which ones should be the parents. The parents consist of male and female individuals. If there is not non-intimate male individual for female then there are no offspring. When creating offspring, the male and female are chosen as parents. Crossover points are chosen randomly [15]. In mutation operation, the mutated individuals are chosen from the new population using equal mutation rates for male and female individuals.

The population of the intermediate population consists of new offspring. Individuals resulted from crossover and mutation operations in the previous generation. The male and female individuals are sorted according to objective function values or fitness values and rejected if individuals have low values.

In infusion operation, if the population is not balanced the new random chromosomes are infused to the next generation. The schematic of the IAMGA procedure is shown in Figure 1.

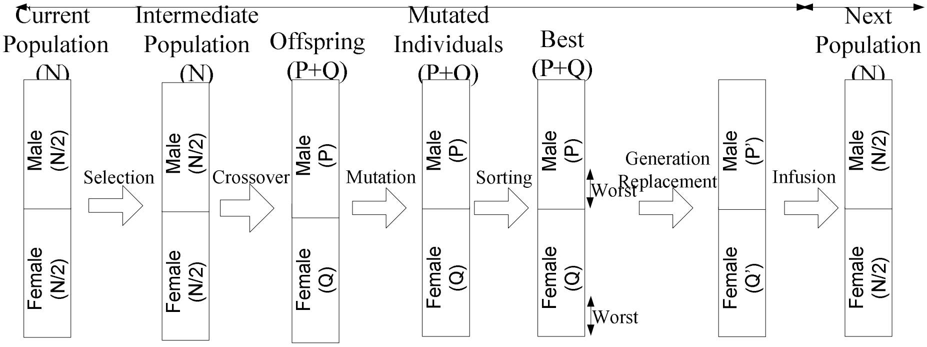


Fig. 1. A Schematic of IAMGA procedure

4.1 Representation

This study provides each individual with an additional feature:

- GenderBit: The sex or gender, as “GenderBit” with male and female values. “GenderBit” does not change in GAs operations such as mutation and crossover
- ID, FatherID and MotherID: unique ID to use for recognize intimate relationships

4.2 Crossover

The main difference among IAMGA and SGA is the method used for parent selection in the crossover operators. In SGAs individuals are mating randomly but in IAMGA with assortative mating two different sex parents are mating with extra condition. Mating requires one parent from each sex and parent should not have intimate relationship.

Females are selected in a sequential fashion without replacement which means that each female will only get to be selected for mating. For all unmated female select a male based on SGA selection operator. This process is repeated until all females have mated.

In two status parents have an intimate relationship. The first state is when selected individual ancestor is parent of current individual, or current individual ancestor is parent of selected individual. The second state is when current individual is ancestor of selected individual or selected individual is ancestor of current individual.

4.3 Infusion

The male and female mating constraint can lead the algorithm to deadlock in the case where the population becomes exclusively comprised of one of these two gender type, such that the reproduction is not possible anymore. This approach prevents these situations by infusion of new chromosome to population. Deadlock occurs when male/female population size is less than half of the initial population size. In deadlock IAMGA randomly infuses new male/female chromosomes to population.

```
Procedure IAMGA {
    t = 0;
    Random Initialize Male and Female Populations(t);
    /*with no preference for any of gender types*/
    Do{
       t = t + 1;
       Selection(t);
       Select_Mothers(t);/*From FemalePopulation(t);*/
       Do{
            Select_Fathers(t);/*From MalePopulation(t);*/
         }while(not intimate relationship between parent)
       Foreach(pair of parents in parents list)   {
         Perform crossover with probability Pc
         InsertNewMale(t);/*Into MalePopulation(t);*/
         InsertNewFemale(t);/*Into FemalePopulation(t)*/
         If (new offsprings are generated)
           Insert_FamilyTree(t);/*New offsprings*/}
       Mutation(t); /*preserving its gender*/
       Replacement(t); /*1-Elitism*/
       If (Male and FemalePopulation(t) are not balanced)
         Infusion(t);
       Delete_FamilyTree(t);/*Old familytree nodes*/
      }While (not termination condition are met)}
```

5 Experimental Results

To prove the concept of IAMGA, simulation studies were performed and applied to solve several test problems. For comparisons, IAMGA was compared with SGA. Selected benchmark functions are shown in Table 1. In Table 2, the proposed algorithm was tested on benchmark functions provided by CEC2010 Special Session on Large Scale Global Optimization [18-19]. Table 3 shows result for benchmark functions. Crossover and mutation methods are uniform and bit flip. Crossover and mutation rates set to 0.75 and 0.05. Replacement is 1-elitism generational method. Selection method is roulette wheel. The algorithm is conducted 20 runs for each test function. The same methods and parameter settings used for SGA and IAMGA. F8's best stability and convergence diagrams are shown in Figure 2.

This approach introduces a gender separation and a special "family tree" structure which indirectly defines intimate relationship among individuals.

Table 1. Benchmark function (F1-F12)

Name	Function	Limits
F1	$\sum_{1}^{3} x_i^2$	$-5.12 \leq x_i \leq 5.12$
F2	$\sum_{1}^{20} x_i^2$	$-5.12 \leq x_i \leq 5.12$
F3	$100(x_1^2 - x_2)^2 + (x_1 - 1)^2$	$-2.048 \leq x_i \leq 2.048$
F4	$\lvert x \rvert + \cos{(x)}$	$-10 \leq x \leq 10$
F5	$\lvert x \rvert + sin\,(x)$	$-10 \leq x \leq 10$
F6	$xsin(4x) + 1.1ysin(2y)$	$0 \leq x, y \leq 10$
F7	$\sum_{i=1}^{2}(x_i^2 - 10\cos(2\pi x_i) + 10)$	$-5.12 \leq x_i \leq 5.12$
F8	$\sum_{i=1}^{20}(x_i^2 - 10\cos(2\pi x_i) + 10)$	$-5.12 \leq x_i \leq 5.12$
F9	$1 + \sum_{i=1}^{2}\left(\frac{x_i^2}{4000}\right) - \prod_{i=1}^{2}\left(\cos\left(\frac{x_i}{\sqrt{i}}\right)\right)$	$-600 \leq x_i \leq 600$
F10	$1 + \sum_{i=1}^{10}\left(\frac{x_i^2}{4000}\right) - \prod_{i=1}^{10}\left(\cos\left(\frac{x_i}{\sqrt{i}}\right)\right)$	$-600 \leq x_i \leq 600$
F11	$2*418.9829 + \sum_{i=1}^{2} -x_i sin\,(\sqrt{\lvert x_i \rvert})$	$-500 \leq x_i \leq 500$
F12	$\sum_{i=1}^{2} 10^{i-1} x_i^2$	$-10 \leq x_i \leq 10$

Table 2. The results achieved by SGA and IAMGA on the test suite: Population size and number of generation set to 40 and 5000

Fun.	SGA		IAMGA	
	Mean	Std.	Mean	Std.
Shifted Ackley	2.01E+01	1.48E-01	**1.72E+01**	2.36E-01
Single-group Shifted and m-rotated Elliptic	4.28E+14	1.56E+14	**1.17E+14**	4.61E+13
Single-group Shifted and m-rotated Rastrigin	5.71E+08	6.57E+07	**3.05E+08**	6.50E+07
Single-group Shifted m-dim. Schwefel	1.38E+11	2.88E+10	**2.91E+10**	5.09E+09
D/m-group Shifted and m-rotated Elliptic	2.26E+10	1.55E+09	**1.90E+10**	1.66E+09
Shifted Rosenbrock	4.61E+11	7.99E+10	**2.38E+11**	4.20E+10

Practicable advantage of using IAMGA approach is using crossover operator with additional condition, which based on intimate relationship. It uses dynamic population and tries to simulate human nature behavior like gender separation, varied size and marriage. They are the most important advantage of IAMGA.

Experiment results show that IAMGA robustly provides better than SGA. The higher performance of IAMGA may be because that this algorithm maintains a higher genetic diversity in the population thus avoiding being trapped in local optima.

Table 3. Comparing IAMGA and SGA

Fun.	Population Size	Number of Generation	Optimum	SGA		IAMGA	
				Mean	Standard Deviation	Mean	Standard Deviation
F1	100	500	0	4.41E-02	5.38E-02	**4.45E-03**	1.34E-02
F2	500	1000	0	1.39E+01	4.34E+00	**7.54E+00**	2.88E+00
F3	80	300	0	1.50E-02	3.58E-03	**6.80E-04**	8.40E-04
F4	20	200	1	1.00E+00	6.50E-03	**1**	0
F5	20	200	0	2.50E-05	6.30E-05	**0**	0
F6	50	200	-18.5547	-18.1385	0.6816	**-18.3335**	0.4838
F7	80	1000	0	3.34E-02	8.32E-02	**3.40E-04**	2.60E-04
F8	100	5000	0	9.92E+01	1.38E+01	**3.61E+01**	1.76E+01
F9	100	800	0	1.60E-03	4.60E-03	**2.80E-04**	1.10E-03
F10	500	2000	0	3.44E+00	2.34E+00	**6.30E-02**	0.1397
F11	100	800	0	2.21E-01	4.09E-01	**4.88E-02**	5.39E-02
F12	50	1000	0	1.69E-02	3.02E-02	**3.50E-04**	1.00E-03

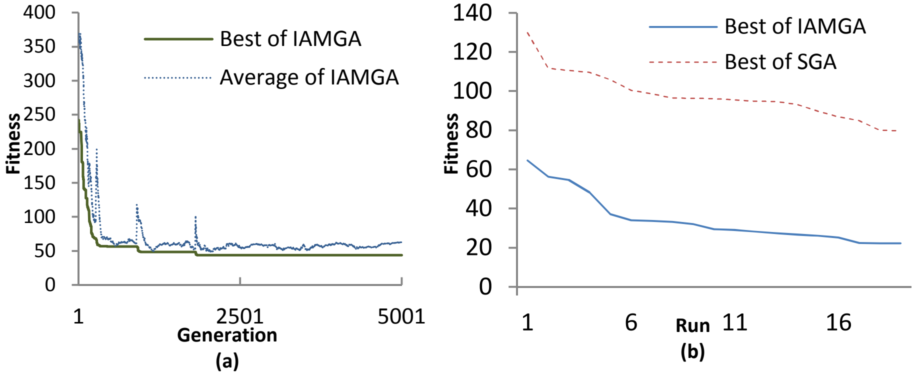


Fig. 2. (a) F8's Convergence Diagram (b) F8's Best Stability Diagram

6 Conclusion and Future Works

The main idea that helps IAMGA to improve its performance is using "Intimate Relationship" because relative individuals have the same ancestors which have many genes in common.

A practical advantage of using a population-based approach is that it can be parallelized naturally. We are currently working on an efficient parallelization of our code which will provide the practical basis for more nature based genetic algorithm. As future work, we are considering to locally improve individuals by purge duplicate individuals and replace them with "productivity".

References

1. Goldberg, D.E.: Genetic algorithms in Search, Optimization and Machine Learning. Addison-Wesley (1989)
2. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor (1975)
3. de Castro, J.P., Posta, A., Bittencourt, G.: A Genetic Algorithm with Feminine Selection. In: Proceedings of the 16th Eureopean Conference on Artificial Intelligence, ECAI 2004, pp. 244–248 (2004)
4. Sanchez-Velazco, J., Bullinaria, J.A.: Sexual Selection with Competitive/Co-Operative Operators for Genetic Algorithms. In: IASTED International Conference on Neural Networks and Computational Intelligence, NCI (2003)
5. Ansótegui, C., Sellmann, M., Tierney, K.: A Gender-Based Genetic Algorithm for the Automatic Configuration of Algorithms. In: Gent, I.P. (ed.) CP 2009. LNCS, vol. 5732, pp. 142–157. Springer, Heidelberg (2009)
6. Savic, A., Tosic, D., Maric, M., Kratica, J.: Genetic Algorithm Approach for Solving the Task Assignment Problem. Serdica Journal of Computing 2(3), 267–276 (2008)
7. Mahfouz, S.Y., Toropov, V.V., Westbrook, R.K.: Modification, tuning and testing of a GA for structural optimization problems. In: 1st AMSO UK/ISSMO Conference on Engineering Design Optimization, pp. 271–278 (1999)
8. Vrajitoru, D.: Simulating gender separation with genetic algorithms. In: Proceedings of the Genetic and Evolutionary Compuatation Conference (GECCO 2002), pp. 634–641 (2002)
9. Lis, J., Eiben, A.E.: A Multi-Sexual Genetic Algorithm for Multiobjective Optimization. In: Proceedings of the 1996 International Conference on Evolutionary Computation, Nagoya, Japan, pp. 59–64. IEEE (1996)
10. Rejeb, J., AbuElhaija, M.: New gender genetic algorithm for solving graph partitioning problems. In: Proceedings of the 43rd IEEE Midwest Symposium on Circuits and Systems, vol. 1, pp. 444–446 (2000)
11. Allenson, R.: Genetic algorithms with gender for multi-function optimization. Technical Report EPCC-SS92-01, Edinburgh Parallel Computing Centre (1992)
12. Ratford, M., Tuson, A., Thompson, H.: Applying Sexual Selection as a Mechanism for Obtaining Multiple Distinct Solutions. Presented at Emerging Technologies 1997 (1997a)
13. Ratford, M., Tuson, A., Thompson, H.: The Single Chromosome's Guide to Dating. In: Third International Conference on Artificial Neural Networks and Genetic Algorithms (1997b)
14. Ronald, E.: When Selection meets Seduction. In: The Sixth International Conference on Genetic Algorithms, pp. 167–173 (1995)
15. Song Goh, K., Lim, A., Rodrigues, B.: Sexual Selection for Genetic Algorithms. Artificial Intelligence Review 19(2), 123–152 (2003)
16. Garcia-Martinez, C., Lozano, M.: Hybrid Real-Coded Genetic Algorithms with Female and Male Differentiation. In: Congress on Evolutionary Computation, pp. 896–903 (2005)
17. Wagner, S., Affenzeller, M.: SexualGA: Gender-Specific Selection for Genetic Algorithms. In: The 9th World Multi-Conference on Systemics, Cybernetics and Informatics, pp. 76–81 (2005)
18. Tang, K.: Benchmark Functions for the CEC 2010 Special Session and Competition on Large-Scale Global Optimization: Nature Inspired Computation and Applications Laboratory (2010), `http://nical.ustc.edu.cn/cec10ss.php` Technical report
19. De Jong, K.A.: Analysis of the Behavior of a Class of Genetic Adaptive Systems, Ph.D. thesis, University of Michigan, Ann Arbor (1975)
20. Randy, L., Haupt, S.: Practical Genetic Algorithms. Wiley-IEEE Publication (2003)

SVR with Chaotic Genetic Algorithm in Taiwanese 3G Phone Demand Forecasting

Li-Yueh Chen[1], Wei-Chiang Hong[2], Bijaya Ketan Panigrahi[3], and Shih Yung Wei[4]

[1] Department of Global Marketing and Logistics, MingDao University
369 Wen-Hua Rd., Peetow, Changhua, 52345, Taiwan, R.O.C.
liyuchen@mdu.edu.tw
[2] Department of Information Management, Oriental Institute of Technology,
58 Sec. 2, Sichuan Rd., Panchiao, Taipei, 220, Taiwan, R.O.C.
samuelsonhong@gmail.com
[3] Department of Electrical Engineering, Indian Institutes of Technology (IITs), India
bkpanigrahi@ee.iitd.ac.in
[4] Department of Information Management, Oriental Institute of Technology,
58 Sec. 2, Sichuan Rd., Panchiao, Taipei, 220, Taiwan, R.O.C.

Abstract. Along with the increases of 3G relevant products and the updating regulations of 3G phones, 3G phones are gradually replacing 2G phones as the mainstream product in Taiwan. Therefore, accurate 3G phones demand forecasting is necessary for those communication related enterprises. Recently, support vector regression (SVR) has been successfully applied to solve nonlinear regression and time series problems. This investigation presents a 3G phones demand forecasting model which combines chaotic sequence with genetic algorithm to improve the forecasting performance. Subsequently, a numerical example of 3G phones demand data from Taiwan is used to illustrate the proposed SVRCGA model. The empirical results reveal that the proposed model outperforms the other three existed models, namely the autoregressive integrated moving average (ARIMA) model, the general regression neural networks (GRNN) model, and SVRGASA model.

Keywords: Chaotic genetic algorithm (CGA), support vector regression (SVR), third generation (3G) phones demand.

1 Introduction

In Taiwan, the establishment of base stations, the updating of regulations governing 3G phones, and the relevant 3G products are also gradually mature, thus, 3G phones are gradually replacing 2G phones as the market's mainstream product. However, for telecommunications businesses, it is important to understand the development tendency of the 3G market and the growth of the 3G phones penetration rate to allocate their investments in base station settings and launched services. Therefore, an accurate forecast of 3G phones demand, usually measured as a number of 3G phones subscribers, is important to help telecommunications companies with making operational, tactical marketing strategic decisions. Conventional quantitative forecasting models are categorized as regression models and time-series models. Regression models, also known as econometric models, are based on traditional

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 248–256, 2011.
© Springer-Verlag Berlin Heidelberg 2011

statistical theory, and focus on identifying relationships among mobile phone subscribers and socio-economic factors, such as income, living expenditure, and same generations' effects. These socio-economic factors are often found out that their coefficients are insignificant; on the contrary, eliminating the many redundant variables may raise the explanation ability (denoted by R^2), but raises the co-linearity problem. This is the major limitations of econometric models [1].

The ARIMA (autoregressive moving integrated moving average) models developed by Box and Jenkins [2], are among the most popular time series forecasting models, and are often employed when data are inadequate to construct econometric models, or when knowledge of the structure of regression models is limited. Time series models are likely to outperform regression models in some cases, such as in short-term forecasting [1,3]. However, a fundamental limitation for time-series forecasting models is their inability to predict changes that are not clear in historical data, particularly for the nonlinearity of mobile phone subscribers' patterns.

Due to significant advances in pattern recognition methodology, recently has enabled the use of artificial neural network (ANN) to forecast business demand. ANN is primarily based on a model of emulating the processing of human neurological system identify related spatial and temporal characteristics from the historical data patterns (particularly for nonlinear and dynamic evolutions). Particularly, ANN does not need to understand any assumptions made by traditional statistical/ econometric approaches. Even though ANN-based forecasting models can approximate to any function, particularly nonlinear functions, ANN models have difficulties in explaining not only black-box operations (such as determining a suitable network structure), but also the non-convex problem of network training errors. Therefore, ANN has difficulty in finding the global optimum [4].

Support vector regression (SVR) [5] has been successfully used to solve forecasting problems in many fields, such as financial time series (stocks index and exchange rate) forecasting [6,7], tourist arrival forecasting [8], atmospheric science forecasting [9,10], traffic flow forecasting [11,12], and electric load [13-15]. Empirical results have demonstrated that the selection of the three parameters (C, ε, and σ) in a SVR model significantly influences the forecasting accuracy. Numerous investigations have given recommendations on appropriate settings of SVR parameters [16], but do not consider all of the interaction effects among the three parameters. No general consensus exists over parameter settings, opinions of scholars differ widely. Therefore, evolutionary algorithms are employed to determine the appropriate parameter values. SVR with genetic algorithm (GA) has been shown to be superior to other competitive forecasting models (ARIMA and ANNs) [8]. However, the population diversity is significantly reduced after some generations, meaning that GA might lead to a premature convergence to a local optimum in the searching for suitable parameters of a SVR model. To overcome these drawbacks, some effective approaches and improvements on GA need to be discovered to maintain the population diversity. One feasible scheme focuses on the chaos approach, due to its easy implementation and unique ability to avoid becoming trapped in local optima [17]. Chaos, defined as highly unstable motion in finite phase space, often occurs in deterministic nonlinear dynamic systems [18]. Such motion is very similar to a random process ("*randomicity*"). Therefore, any variable in the chaotic space can travel ergodically over the whole space of interest ("*ergodicity*"). Additionally, it is extremely sensitive to the initial condition, which is an important property sometimes referred to as the so-called butterfly effect [19]. Based on these advantages of the chaos, the chaotic optimization algorithm (COA) was proposed to solve complex function optimization [18]. Yuan et al. [20] recently presented the chaotic genetic algorithm (CGA), which integrates GA with chaotic mapping operator, to exploit the respective searching advantages of both models.

This investigation aims to solve this problem of maintaining the population diversity of GA in determining the three free parameters in the SVR forecasting model. Therefore, the proposed SVR model, namely SVRCGA, employs the CGA method proposed by Yuan et al. [20], to provide good forecasting performance in capturing 3G phones demand changes tendency. The remainder of this paper is organized as follows. The SVRCGA model is introduced in section 2, including the formulation of SVR and the CGA. Numerical examples to demonstrate the forecasting performance of the proposed model and corresponding comparison results with the other forecasting models are provided in Section 3. Conclusions are finally made in Section 4.

2 Methodology of SVRCGA Model

2.1 Support Vector Regression

The brief ideas of SVMs for the case of regression are introduced. A nonlinear mapping $\varphi(\cdot)$: $\Re^n \rightarrow \Re^{n_h}$ is defined to map the input data (training data set) $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ into a so-called high dimensional feature space (which may have infinite dimensions), $\Re^{n_h}$. Then, in the high dimensional feature space, there theoretically exists a linear function, f, to formulate the nonlinear relationship between input data and output data. Such a linear function, namely SVR function, is as Eq. (1),

$$f(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\varphi(\mathbf{x}) + b \quad (1)$$

where $f(\mathbf{x})$ denotes the forecasting values; the coefficients $\mathbf{w}$ ($\mathbf{w} \in \Re^{n_h}$) and b ($b \in \Re$) are adjustable. As mentioned above, SVM method aims at minimizing the empirical risk,

$$R_{emp}(f) = \frac{1}{N}\sum_{i=1}^{N}\Theta_\varepsilon(y_i, \mathbf{w}^T\varphi(\mathbf{x}_i) + b). \quad (2)$$

where $\Theta_\varepsilon(\mathbf{y}, f(\mathbf{x}))$ is the ε-insensitive loss function and defined as Eq. (3),

$$\Theta_\varepsilon(\mathbf{y}, f(\mathbf{x})) = \begin{cases} |f(\mathbf{x}) - \mathbf{y}| - \varepsilon, & if \quad |f(\mathbf{x}) - \mathbf{y}| \geq \varepsilon \\ 0, & otherwise \end{cases}. \quad (3)$$

In addition, $\Theta_\varepsilon(\mathbf{y}, f(\mathbf{x}))$ is employed to find out an optimum hyper plane on the high dimensional feature space to maximize the distance separating the training data into two subsets. Thus, the SVR focuses on finding the optimum hyper plane and minimizing the training error between the training data and the ε-insensitive loss function. Then, the SVR minimizes the overall errors,

$$\underset{\mathbf{w},b,\xi^*,\xi}{\mathrm{Min}} \quad R_\varepsilon(\mathbf{w},\xi^*,\xi) = \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C\sum_{i=1}^{N}(\xi_i^* + \xi_i) \tag{4}$$

with the constraints

$$\begin{gathered} \mathbf{y}_i - \mathbf{w}^{\mathrm{T}}\varphi(\mathbf{x}_i) - b \le \varepsilon + \xi_i^*, \quad i = 1,2,...,N \\ -\mathbf{y}_i + \mathbf{w}^{\mathrm{T}}\varphi(\mathbf{x}_i) + b \le \varepsilon + \xi_i, \quad i = 1,2,...,N \\ \xi_i^* \ge 0, \ i = 1,2,...,N \\ \xi_i \ge 0, \ i = 1,2,...,N \end{gathered}$$

The first term of Eq. (4), employed the concept of maximizing the distance of two separated training data, is used to regularize weight sizes, to penalize large weights, and to maintain regression function flatness. The second term penalizes training errors of $f(\mathbf{x})$ and $\mathbf{y}$ by using the ε-insensitive loss function. C is a parameter to trade off these two terms. Training errors above ε are denoted as ξ_i^*, whereas training errors below $-\varepsilon$ are denoted as ξ_i.

After the quadratic optimization problem with inequality constraints is solved, the parameter vector **w** in Eq. (1) is obtained,

$$\mathbf{w} = \sum_{i=1}^{N}\left(\beta_i^* - \beta_i\right)\varphi(\mathbf{x}_i). \tag{5}$$

where β_i^*, β_i are obtained by solving a quadratic program and are the Lagrangian multipliers. Finally, the SVR regression function is obtained as Eq. (6) in the dual space,

$$f(\mathbf{x}) = \sum_{i=1}^{N}\left(\beta_i^* - \beta_i\right)K(\mathbf{x}_i,\mathbf{x}) + b \tag{6}$$

where $K(\mathbf{x}_i,\mathbf{x}_j)$ is called the kernel function, and the value of the kernel equals the inner product of two vectors, $\mathbf{x}_i$ and $\mathbf{x}_j$, in the feature space $\varphi(\mathbf{x}_i)$ and $\varphi(\mathbf{x}_j)$, respectively; that is, $K(\mathbf{x}_i,\mathbf{x}_j) = \varphi(\mathbf{x}_i) \circ \varphi(\mathbf{x}_j)$. Any function that meets Mercer's condition can be used as the kernel function. The Gaussian RBF kernel function is specified in this study.

It is well known that the forecasting accuracy of a SVR model depends on a good setting of hyper parameters C, ε, and the kernel parameters (σ). Thus, the determination of all three parameter selection is further an important issue. There is no structural method or any shortage opinions on efficient setting of SVR parameters. In addition, aforementioned, GA is lack of knowledge memory functions, which leads to time consuming and a premature convergence to a local optimum in the searching the suitable parameters of a SVR model. Therefore, the CGA is used in the proposed SVR model to optimize the parameter selection.

2.2 Chaotic Genetic Algorithm in Selecting Parameters of the SVR Model

Traditional GA and related improvements have a common arrangement that completely ignore the individuals' experiences during their lifetime. This is because they are based on randomized searches, with no necessary connections between the current and next generations except for some controlling operators such as crossover and mutation operators. Mutation can continuously pursue individuals of higher fitness value, and guide the evolution of the whole population. In contrast, if the precision is satisfactory, then the solution is often trapped in the local optima, or takes too long to converge. Therefore, this investigation applies the annealing chaotic mutation operation, proposed by Cheng et al. [21]. The annealing chaotic mutation operation not only simulates the chaotic evolutionary process of biology, but also employs chaotic variable to perform ergodic searching of the solution space, enabling it to find another better solution in the current neighborhood area of optimum solution, and to let GA possess ongoing motivity all along. The process of the proposed procedure of CGA is as follows.

Step 1: Generate initial population by chaotic mapping operator (CMO). The values of the three parameters in a SVR model in the *i*th iteration can be represented as $X_k^{(i)}, k = C, \sigma, \varepsilon$. Set $i = 0$, and employ Eq. (7) to map the three parameters among the intervals (Min_k, Max_k) into chaotic variable $x_k^{(i)}$ located in the interval (0, 1).

$$x_k^{(i)} = \frac{X_k^{(i)} - Min_k}{Max_k - Min_k}, \quad k = C, \sigma, \varepsilon \qquad (7)$$

Then, adopt Eq. (8) with $\mu = 4$ to compute the next iteration chaotic variable, $x_k^{(i+1)}$. Transform $x_k^{(i+1)}$ to obtain three parameters for the next iteration, $X_k^{(i+1)}$, by the following Eq. (9).

$$\begin{aligned} x_k^{(i+1)} &= \mu x_k^{(i)} (1 - x_k^{(i)}) \\ x_k^{(i)} &\in (0,1), \quad i = 0, 1, 2, \ldots \end{aligned} \qquad (8)$$

$$X_k^{(i+1)} = Min_k + x_k^{(i+1)} \left(Max_k - Min_k \right) \qquad (9)$$

After this transformation, the three parameters, C, σ , and ε , are encoded into a binary format, and are represented by a chromosome composed of “genes” of binary numbers. Each chromosome has three genes, respectively representing three parameters.

Step 2: Evaluate fitness. Evaluate the fitness (forecasting errors) of each chromosome. In this work, a negative mean absolute percentage error (-MAPE) is used as the fitness function. The MAPE is calculated as Eq. (10),

$$MAPE = \frac{1}{N} \sum_{i=1}^{N} \left| \frac{y_i - f_i}{y_i} \right| \times 100\% . \qquad (10)$$

where y_i and f_i represent the actual and forecast values, and N is the number of forecasting periods.

Step 3: Selection. Based on fitness functions, chromosomes with higher fitness values are more likely to yield offspring in the next generation. The roulette wheel selection principle is applied to choose chromosomes for reproduction.

Step 4: Crossover operations. In crossovers, chromosomes are paired randomly. The proposed scheme adopts the single-point-crossover principle. Segments of paired chromosomes between two determined break-points are swapped. Finally, the three crossover parameters are decoded into a decimal format.

Step 5: Annealing chaotic mutation. For the *i*th iteration (generation) crossover population ($\hat{X}_k^{(i)}, k = C, \sigma, \varepsilon$) of current solution space (Min_k, Max_k) are mapped to chaotic variable interval [0,1] to form the crossover chaotic variable space $\hat{x}_k^{(i)}, k = C, \sigma, \varepsilon$, as Eq. (11),

$$\hat{x}_k^{(i)} = \frac{\hat{X}_k^{(i)} - Min_k}{Max_k - Min_k}, k = \mathrm{C}, \sigma, \varepsilon, i = 1, 2, \ldots\ldots, q_{\max} \quad (11)$$

where $q_{\max}$ is the maximum evolutional generation of the population. Then, the *i*th chaotic variable $x_k^{(i)}$ is summed up to $\hat{x}_k^{(i)}$ and the chaotic mutation variable are also mapped to interval [0, 1] as in Eq. (12),

$$\tilde{x}_k^{(i)} = \hat{x}_k^{(i)} + \delta x_k^{(i)} . \quad (12)$$

where δ is the annealing operation. Finally, the chaotic mutation variable obtained in interval [0, 1] is mapped to the solution interval (Min_k, Max_k) by definite probability of mutation (p_m), thus completing a mutative operation.

$$\tilde{X}_k^{(i)} = Min_k + \tilde{x}_k^{(i)} \left(Max_k - Min_k\right) . \quad (13)$$

Step 6: Stop condition. If the number of generations is equal to a given scale, then the best chromosomes are presented as a solution; otherwise, go back to Step 2.

3 Numerical Examples

3.1 Data Set

This study uses historical monthly 3G mobile phones demand data (2006 to 2008), obtained from the revenue reports section of Chunghwa Telecom Co. Ltd. [22], to compare the forecasting performance of the proposed SVRCGASA model with those of ARIMA, GRNN, and SVRGASA models proposed by Hong et al. [23]. To be based on the same compared condition, the 3G mobile phone demand data is also divided into the three periods, i.e., training period (January 2006 to May 2007), validation period (June 2007 to October 2007), and testing period (November 2007 to March 2008), accordingly.

3.2 Parameters Determination of the SVRCGA Model

The parameters of the CGA algorithm are experimentally set as follows, the population size (p_{size}) is 200; the probability of crossover (p_c) is 0.5; the probability of mutation (p_m) is 0.1; the annealing operation parameter (δ) is 0.9. In the training stage, the rolling-based forecasting procedure is conducted, which dividing training data into two subsets, namely fed-in subset and fed-out subset, respectively. While training errors improvement occurs, the three kernel parameters, σ, C, and ε of the SVRCGA model adjusted by CGA are employed to calculate the validation error. Then, the adjusted parameters with minimum validation error are selected as the most appropriate parameters. Finally, a one-step-ahead policy is employed to forecast 3G phone demand. The forecasting results and the suitable parameters for the different SVRCGA models are illustrated in Table 1, in which it is indicated that the SVRCGA model performs the best when 13 fed-in data are used. Table 2 shows the actual values and the forecast values (from Nov. 2007 to Mar. 2008) obtained from various forecasting models. The proposed SVRCGA model has smaller MAPE values than other alternatives to capture the 3G demand patterns on monthly average basis. It is mainly caused of the chaotic mapping operator and chaotic mutation process of CGA's searching capability to determine proper parameters combination. For example, in Table 1, the CGA is then excellently to shift the local solution of SVRGA model by 13 fed-in data rolling type, (σ, C, ε)=(0.3959, 1492.0, 54.885) with local optimal forecasting errors, in terms of MAPE (3.108%), to be improved by CGA to another better solution, (σ, C, ε)=(1.1694, 10932.0, 7.0822) to be the appropriate local optimal forecasting error in terms of MAPE (0.577%). Thus, it once again reveals that chaotic sequence could significantly improve the performance in terms of premature convergence.

Table 1. Forecasting results of SVRGA and SVRCGA models

SVRGA [23]					SVRCGA				
Number of fed-in	Parameters			Testing MAPE	Number of fed-in	Parameters			Testing MAPE
	σ	C	ε			σ	C	ε	
10	**0.7411**	**3057.6**	**5.143**	**2.142**	10	0.7925	4155.6	1.0147	1.179
11	0.7722	1005.9	47.980	3.506	11	0.7532	4261.2	1.9456	1.002
12	3.9378	842.88	53.250	3.614	12	0.6304	3890.1	1.2228	1.168
13	0.3959	1492.0	54.855	3.108	**13**	**1.1694**	**10932**	**7.0822**	**0.577**
14	3.4412	1431.1	54.642	2.992	14	1.1251	10230	4.9627	1.124

Table 2. Forecasting Taiwan 3G demand from Nov. 2007 to Mar. 2008

Months Years	Actual demand	ARIMA	GRNN	SVRGA	SVRGASA	SVRCGA
Nov. 2007	2,195	2,226	2,111	2,143	2,207	2,199
Dec. 2007	2,291	2,333	2,195	2,259	2,281	2,289
Jan. 2008	2,365	2,439	2,291	2,353	2,358	2,367
Feb. 2008	2,478	2,544	2,365	2,426	2,449	2,456
Mar. 2008	2,588	2,650	2,478	2,476	2,559	2,546
MAPE		2.287%	3.991%	2.142%	0.720%	0.577%

4 Conclusion

As strong development tendency of 3G related infrastructures, governing regulations, and relevant products, 3G telecommunication is currently going to its age of modern high speed data communications in Taiwan. Therefore, accurate 3G mobile phone demand forecasting would not only well investigate the future development trends of 3G, but also would provide important guide for effective implementations of 3G related businesses nurturing. In this investigation, SVRCGA model was proposed to predict Taiwan 3G phone demand. This study evaluated the feasibility of CGA in parameter determination to achieve forecasting accuracy improvement by integrated into the SVR model. Meanwhile, this research also indicated the chaotic sequence was useful to improve the forecasting accuracy by avoiding being trapped in local optimum like SVRGA did. In the future, other advanced optimization algorithms for parameters selection and other novel hybrid mechanism should be further applied to achieve more accurate demand forecasting. Other socio-economic factors, such as market prices and gross domestic expenditure per person, could be included in the SVRCGA model.

Acknowledgments. This research was conducted with the support of National Science Council, Taiwan, ROC (NSC 100-2628-H-161-001-MY4, NSC 100-2811-H-161-001).

References

1. Sheldon, P.J., Var, T.: Tourism forecasting: A review of empirical research. Journal of Forecasting 4, 183–195 (1985)
2. Box, G.E.P., Jenkins, G.M.: Time Series Analysis: Forecasting and Control. Holden-Day, San Francisco (1976)
3. Witt, S.F., Witt, C.A.: Modeling and Forecasting Demand in Tourism. Academic Press, London (1992)
4. Suykens, J.A.K.: Nonlinear modelling and support vector machines. In: Proceedings of IEEE Instrumentation and Measurement Technology Conference, pp. 287–294 (2001)
5. Vapnik, V., Golowich, S., Smola, A.: Support vector machine for function approximation, regression estimation, and signal processing. Advances in Neural Information Processing Systems 9, 281–287 (1996)
6. Hung, W.M., Hong, W.C.: Application of SVR with improved ant colony optimization algorithms in exchange rate forecasting. Control and Cybernetics 38, 863–891 (2009)
7. Pai, P.F., Lin, C.S., Hong, W.C., Chen, C.T.: A hybrid support vector machine regression for exchange rate prediction. International Journal of Information and Management Sciences 17, 19–32 (2006)
8. Pai, P.F., Hong, W.C., Chang, P.T., Chen, C.T.: The application of support vector machines to forecast tourist arrivals in Barbados: An empirical study. International Journal of Management 23, 375–385 (2006)
9. Hong, W.C.: Rainfall forecasting by technological machine learning models. Applied Mathematics and Computation 200, 41–57 (2008)

10. Pai, P.F., Hong, W.C.: A recurrent support vector regression model in rainfall forecasting. Hydrological Processes 21, 819–827 (2007)
11. Hong, W.C.: Traffic flow forecasting by seasonal SVR with chaotic simulated annealing algorithm. Neurocomputing 74, 2096–2107 (2011)
12. Hong, W.C., Dong, Y., Zheng, F., Wei, S.Y.: Hybrid evolutionary algorithms in a SVR traffic flow forecasting model. Applied Mathematics and Computation 217, 6733–6747 (2011)
13. Hong, W.C.: Hybrid evolutionary algorithms in a SVR-based electric load forecasting model. International Journal of Electrical Power and Energy Systems 31, 409–417 (2009)
14. Hong, W.C.: Application of chaotic ant swarm optimization in electric load forecasting. Energy Policy 38, 5830–5839 (2010)
15. Hong, W.C.: Chaotic particle swarm optimization algorithm in a support vector regression electric load forecasting model. Energy Conversion and Management 50, 105–117 (2009)
16. Cherkassky, V., Ma, Y.: Practical selection of SVM parameters and noise estimation for SVM regression. Neural Networks 17, 113–126 (2004)
17. Wang, L., Zheng, D.Z., Lin, Q.S.: Survey on chaotic optimization methods. Computing Technology and Automation 20, 1–5 (2000)
18. Li, B., Jiang, W.: Optimizing complex functions by chaos search. Cybernetics and Systems 29, 409–419 (1998)
19. Lorenz, E.N.: Deterministic nonperiodic flow. Journal of The Atmospheric Sciences 20, 130–141 (1963)
20. Yuan, X., Yuan, Y., Zhang, Y.: A hybrid chaotic genetic algorithm for short-term hydro system scheduling. Mathematics and Computers in Simulation 59, 319–327 (2002)
21. Cheng, C.T., Wang, W.C., Xu, D.M., Chau, K.W.: Optimizing hydropower reservoir operation using hybrid genetic algorithm and chaos. Water Resources Management 22, 895–909 (2008)
22. Chunghwa Telecom Co. Ltd. Monthly Revenue Reports. Financial Information Service (2008), `http://www.cht.com.tw/CompanyCat.php?Page=FileDownload&CatID=798`
23. Hong, W.C., Dong, Y., Chen, L.Y., Lai, C.Y.: Taiwanese 3G mobile phone demand forecasting by SVR with hybrid evolutionary algorithms. Expert Systems with Applications 37, 4452–4462 (2010)

Genetic Algorithm Assisted Enhancement in Pattern Recognition Efficiency of Radial Basis Neural Network

Prabha Verma and R.D.S. Yadava

Department of Physics, Banaras Hindu University,
Varanasi 221005, India
`{pverma.bhu,ardius}@gmail.com`

Abstract. The paper presents a feature extraction method for improving pattern classification efficiency of the radial basis function neural network. The principal component analysis in combination with preprocessing by vector autoscaling and dimensional autoscaling has been used to generate two alternate feature vector representations of the objects. A feature fusion scheme is proposed in which the two feature sets are combined by simple concatenation and then allowed to undergo genetic evolution. The fused features are obtained by applying a weighting method based on the prevalence of feature components in the terminal population. The present method of feature extraction in combination with radial basis neural network has been demonstrated to improve the classification rate for nine benchmark datasets analyzed.

Keywords: Genetic algorithm, radial basis function neural network, feature extraction, pattern recognition.

1 Introduction

Pattern recognition from a given multivariate data is accomplished in three successive steps: preprocessing, feature extraction and classification. Data preprocessing is to prepare data through proper scaling, normalization, noise or outlier removal to facilitate efficient feature extraction. Feature extraction models preprocessed data in terms of new variables that segregates different object types as distinctly as possible. This is achieved through some linear or nonlinear transformation of measurement space under certain assumptions about multivariate statistics. Classification is to map feature space into object identity space [1]. Several algorithms have been developed for processing data at each stage [2], [3], [4]. A specific pattern recognition task selects some combination of these algorithms that produces most accurate classification result [4]. Usually different combinations work best in different domains of applications. In applications where a set of sensors acquire information of varied nature about target objects an object is represented by the set of sensors output. Interpreting sensors as variables of a multidimensional space the set of sensors output

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 257–264, 2011.
© Springer-Verlag Berlin Heidelberg 2011

represent a vector called pattern vector in measurement space. The feature extraction transforms measurement space into feature space where the objects are represented by feature vectors. The feature components are the object descriptors. Different object types or classes occupy separate regions in the feature space. An efficient feature extraction procedure is that which produces maximally compact class regions with largest possible separation between classes [1, pp. 179-181]. An efficient feature extraction process reduces the complexity at classification stage, and makes the classifier more accurate and robust.

This paper is concerned about enhancing performance of a radial basis function (RBF) neural network classifier by improvising the data representation at its input. The strategy consists of feature extraction by principal component analysis (PCA) in combination with two alternate methods of data preprocessing (vector autoscaling and dimensional autoscaling [5]) followed by genetic algorithm assisted feature fusion. The method is an improvisation over an earlier published work by the authors' group [6], and is described in Section 2. The RBF network is described in Section 3 and validation results are presented in Section 4 where a comparison with the results obtained from other methods is also presented. The paper completes with Section 5 on conclusion.

2 Feature Extraction

The schematic of feature extraction procedure is shown Fig. 1 below. Let the input data consists of M measurement vectors of N dimensions. It can be written as $M \times N$ data matrix $X = (X_1, X_2, X_3, ..., X_M)^T \equiv \{x_{ij}\}$ with $i = 1,2,...,M$, $j = 1,2,...,N$. The matrix elements x_{ij} denote sensors output and $X_i = (x_{i1}, x_{i2}, x_{i3}, ..., x_{iN})$ denotes i-th sample vector. The data is processed by two feature extractors: (i) preprocessing by vector autoscaling and principal component analysis yielding feature vectors $Y_i^1 = (y_{i1}^1, y_{i2}^1, y_{i3}^1, ..., y_{iN}^1)$, and (ii) preprocessing by dimensional autoscaling and principal component analysis yielding feature vectors $Y_i^2 = (y_{i1}^2, y_{i2}^2, y_{i3}^2, ..., y_{iN}^2)$ in N-dimensional feature space defined by eigenvector directions. The vector autoscaling [5] is data mean-centering and variance normalization with respect to measurement variables for each sample vector given as $x_{ij} \leftarrow \dfrac{x_{ij} - \bar{x}_i}{\sigma_i}$ with mean $\bar{x}_i = \dfrac{1}{N}\sum_{j=1}^{N} x_{ij}$ and variance $\sigma_i = \sqrt{\dfrac{1}{N}\sum_{j=1}^{N}(x_{ij} - \bar{x}_i)^2}$. The dimensional autoscaling [5] is data mean-centering and variance normalization with respect to samples for each variable given as $x_{ij} \leftarrow \dfrac{x_{ij} - \bar{x}_j}{\sigma_j}$ with mean $\bar{x}_j = \dfrac{1}{M}\sum_{i=1}^{M} x_{ij}$ and variance $\sigma_j = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(x_{ij} - \bar{x}_j)^2}$. Next, the two feature vectors are combined by simple concatenation of the feature components into a single $2N$-dimensional vector $Z_i = \left(y_{i1}^1, y_{i2}^1, \ldots, y_{iN}^1, y_{i1}^2, y_{i2}^2, \ldots, y_{iN}^2\right)$. The variances of these feature components are renormalized according to $\sigma_k^2 = \sigma_k^2 \Big/ (2N)^{-1}\sum_{l=1}^{2N}\sigma_l^2$ and a new combined feature vector is obtained. Let it be denoted as $Z_i = \left(z_{i1}, z_{i2}, z_{i3}, \ldots, z_{iK}\right)$ with $K = 2N$. This vector Z_i is then allowed to undergo genetic evolution to obtain the final fused feature vector for RBF input.

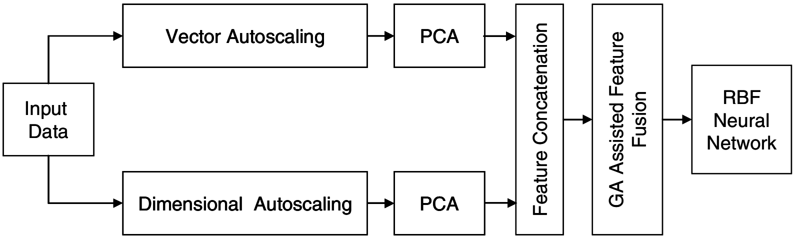


Fig. 1. Schematic of genetic algorithm assisted feature generation and RBF network classifier

2.1 Feature Modification by Genetic Evolution

An arbitrary combined feature vector Z_i in the dataset is made to go through generations of genetic evolution until modifications in feature components stabilized. Formulation of this as genetic algorithm and a method for calculation of feature modifications were presented in [6]. For any Z_i , individual feature components were interpreted as genes, and collection of all feature components as a pool of genes from which an initial population of chromosomes were created. The variances associated with different feature components (genes) were interpreted to represent their significance for object discrimination. The sum of variances of all genes in a chromosome was interpreted to represent its fitness. The methods for creating initial population and evolution through selection, crossover and mutation operations are briefly as follows.

Initial Population. A fixed size initial population of chromosomes was created by thresholding probability distance matrix calculated according to probability distance metric introduced by Zohdy et al. [7], [8]. For a combined feature vector $Z_i = \left(z_{i1} \ldots z_{ik} \ldots z_{il} \ldots z_{iK}\right)$, if it is assumed that k-th feature component is fully accurate, then how accurate l-th component is, is given by the probability distance defined as

$d_{i,kl} = \frac{1}{\sigma_k\sqrt{2\pi}} erf\left(\frac{z_{il} - z_{ik}}{\sigma_k\sqrt{2}}\right)$ where σ_k is the normalized variance of the *k*-th feature component as defined above, and *'erf'* denotes the error function. By assuming one by one in turn that each feature component in Z_i is fully accurate, we obtain $K \times K$ probability distance matrix $D_i = \{d_{i,kl}\}$ with $k = 1,...,K$, $l = 1,...,K$. The meaning of probability distance is that if a feature component is assumed to be fully accurate then any other component which is closer would be more accurate than others. The diagonal matrix elements $d_{i,kl}(k = l)$, being the distance of a component from itself, are all zero because $erf(0) = 0$. To define a chromosome we apply a threshold on probability distance d_c such that the feature components away from this distance are not significant, and are set to become zero. Applying this criterion for each row of the D_i matrix an initial population of K chromosomes is generated. The members of initial population were then ranked according to the values of their fitness (sum of variances).

Genetic Evolution. The standard genetic algorithm has been implemented by using the function '*gaoptimset'* available in the Matlab GA toolbox. A genetic algorithm was created to produce three types of children for the next generation – elite, crossover and mutation. The numbers of children in each category were adjusted by choosing the function parameters 'elite count', 'crossover fraction' and 'mutation rate'. The next generation population is obtained as, *next generation = elite children + crossover children + mutation children.* A fix size population was maintained at each generation. The evolutionary process was continued until the average fitness of the current population stabilizes. The average fitness of a generation is calculated by using the normalized variances of the gene pool. The sum of gene (feature) variances for any member (chromosome) in current generation is its fitness. The average of this sum-variance over the entire population is defines the average fitness of the current generation. This average generation fitness is monitored at successive generations. The number of generations needed for reaching fitness stability varies from data to data. Fig. 2 shows an example of the fitness convergence while doing the analysis of the glass data (to be discussed below).

Feature Modification. Finally, the modified feature vector was obtained by weighting the feature components of the combined feature vector $Z_i = \{z_{ij}\} = [z_{i1}\ z_{i2}\ z_{i3} .... z_{iK}]$ according to the prevalence of respective feature components in the GA produced terminal population. Drawing parallel from the Shannon's entropy in information theory [9], we assumed that the feature components with higher values of Shannon's entropy carry more information, hence are important for object description. If *j*-th feature component has appeared n_j times in the terminal population then the probability of its occurrence is $p_j = n_j / K^2$ where K^2 denotes the population size because there are K chromosomes of length K in the fixed size population at every generation. The Shannon'n entropy (average information content) associated with this component is given by $-p_j \log_2 p_j$ in bits. We used the feature weighting defined as $z_{ij} = z_{ij}\left(1 - p_j \log_2 p_j\right)$. In earlier work [6] the weight factor had been taken as $z_{ij} = z_{ij}\left(1 + p_j\right)$.

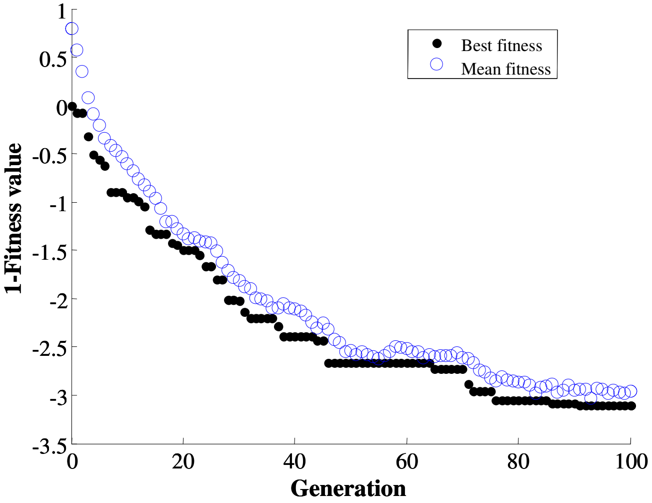


Fig. 2. Convergence of average population fitness with successive generations of genetic evolution in glass data analysis

3 RBF Neural Network Classifier

The radial basis function (RBF) neural network architecture used for classification is a three layer (input, hidden and output) network, and is implemented using '*newrb*' function in Matlab. The input layer defined by the dimensionalities of the feature vectors. The hidden layer consists of radial basis neurons '*radbas*' with Gaussian activation $a = \exp(-x^2)$ producing numeric output over [0, 1]. The input to radial basis transfer function is calculated as $x = \|\mathrm{w} - \mathrm{Z_i}\| b$ where $\|\ \|$ denotes the vector distance (Euclidian norm) between weight vector w and input vector $\mathrm{Z_i}$. The output layer uses a linear transfer function '*purelin*' which is scaled and normalized to yield the final output over [0, 1]. The class prediction is done on the basis of maximum output.

4 Validation Results

In the '*gaoptimset*' implementation of genetic algorithm the selection, crossover and mutation operations were implemented using the functions 's*electionremainder*', '*crossoverscattered*' with crossover fraction 0.9 and '*mutationuniform*' with mutation rate 0.2. All the unspecified parameters were set to their default values. The random number seed generation in the genetic algorithm was initialized at its default value. The validation analysis has been carried out on 9 datasets. The 4 dataset represented odor analysis by sensor array [10-12] and 5 were benchmark datasets from other domains available at UCI machine learning repository [12]. These datasets were selected to challenge the algorithm at varied levels of difficulty. A comparison of RBF generated results was carried out for to kinds of inputs (with and without GA assisted fusion). A summary of datasets is given in Table 1 and the classification results are presented in Table 2. The results in column 3 of Table 2 are from the present GA assisted RBF classifier. It can be that these results are the best among all the tree variants of feature extraction and classifier combination studied. Also, the results are better than some of best results reported in literature as shown in the last column of this table.

Table 1. Summary of datasets used in the present analysis

Dataset	Classes	Variables	Samples	Remark
Coffee (blend)	7	5	249	The classes are coffee aroma of blend and monovariety groups [10].
Coffee (monovariety)	7	4	210	-
Nerve agents	2	3	125	The classes are nerve agent and non-nerve agent gases [11].
Wine	3	13	178	The classes are wines derived from three different cultivars [12].
Iris	3	4	150	The classes are different types of iris plants [12].
Glass Identification	6	9	214	The classes are types of glasses of forensic interest [12].
Pima-Indian (diabetes)	2	8	768	The classes represent whether the patients tested positive or negative [12].
Haberman (breast cancer survival)	2	3	306	The classes are survival status of the patients who had undergone surgery for breast cancer [12].
Auto-mpg (automobile)	3	8	398	The classes are city-cycle fuel consumption [12].

Table 2. Classification results using GA assisted feature fusion and RBF network

Datasets	% Classification Rate			Some Results Reported by Others
	RBF without GA	GA assisted RBF with $z_{ij} = z_{ij}(1 - p_j \log_2 p_j)$	GA assisted BPNN with $z_{ij} = z_{ij}(1 + p_j)$	
Coffee (blend)	90.48	96.83	87.3	87% MLP [10]
Coffee (mono)	76.78	85.71	78.57	82% MLP [10]
Nerve Agents	90.32	98.06	96.0	97% ANN [13]
Wine	97.26	98.63	98.6	97.8% FDA [14]; 89.3 - 97.2% KNNBA [15]
Iris	96.71	98.33	98.3	98.1% FDA [14]
Glass Identification	65.52	75.86	71.3	67.1% FDA [14]; 48.6-68.6% KNNBA [15]
Pima-Indian (diabetes)	66.67	83.33	81.0	76% MLP [16]
Haberman (breast-cancer survival)	73.33	80	75	73% MLP [16]; 59.2-73.3% KNNBA [15]
Auto-mpg (automobile)	74.24	86.00	82.0	79.1% ODT[17]

MLP: Multilayer Perceptron, BPNN: Backpropagation Neural Network
ANN: Artificial Neural Network, FDA: Fisher Discriminant Analysis
KNNBA: K-nearest neighbor based association, ODT: Ordinal Decision Tree

5 Conclusion

The data prepared by two preprocessing methods (vector autoscaling and dimensional autoscaling) presents data from different perspectives to the PCA processor. The search by PCA algorithm for orthogonal maximum variance directions in the data space is therefore expected to bring out complimentary information about the objects. The combined feature vector Z can thus be expected to give richer representation. The genetic evolution of these feature vectors is expected to enhance object representation further by adding extra weights on the feature components of higher significance. The RBF network with richer input can be thus expected to yield better results. The results obtained by this method are highly repeatable unlike the backpropagation neural network studied earlier, column 4 in Table 2. The multiple preprocessor and GA assisted feature extraction method in combination with RBF neural network classifier proposed in this paper enhances classification efficiency for all datasets analyzed here compared to the most published reports on the same datasets by other (often more complex) methods.

Acknowledgments. This work is supported by the Government of India, Defence Research & Development Organization Grant No. ERIP-ER-0703643-01-1025. The authors are thankful to all contributors whose data were used for validation work in this paper.

References

1. Theodoridis, S., Koutroumbas, K.: Pattern Recognition, ch. 1. Academic, San Diego (2003)
2. Kotsiantis, S.B.: Supervised Machine Learning: A Review of Classification Techniques. Informatica 31, 249–268 (2007)
3. Jain, A.K.: Statistical Pattern Recognition: A Review. IEEE Trans. Pattern Anal Machine Intelligence 22(1), 4–37 (2000)
4. Gutierrez-Osuna, R.: Pattern Analysis for Machine Olfaction: A Review. IEEE Sensors J. 2(3), 189–202 (2002)
5. Osuna, R.G., Nagle, H.T.: A Method for Evaluating Data Preprocessing Techniques for Odor Classification with an Array of Gas Sensors. IEEE Trans. Syst. Man Cybern. B 29(5), 626–632 (1999)
6. Somvanshi, D., Yadava, R.D.S.: Boosting Principal Component Analysis by Genetic Algorithm. Defence Science Journal 60, 392–398 (2010)
7. Zohdy, M.A., Loh, N., Liu, J.: Application of Maximum Likelihood Identification with Multisensor Fusion to Stochastic Systems. In: Proceedings of American Control Conference, Pennsylvania, pp. 411–416 (1989)
8. Khan, A.A., Zohdy, M.A.: A Genetic Algorithm for Selection of Noisy Sensor Data in Multisensor Data Fusion. In: Proceedings of American Control Conference, Albuquerque, pp. 2256–2262 (1997)
9. Cover, T.M., Thomas, J.A.: Elements of Information Theory, Section 2.1. Wiley & Sons, Hoboken (2006)
10. Pardo, M., Sberveglieri, G.: Coffee Analysis with an Electronic Nose. IEEE Trans. Instrum. Measur. 51, 1334–1339 (2002), `http://sensor.ing.unibs.it/_people/pardo/dataset.html`
11. Rose-Pehrson, S.L., Lella, D.D., Grate, J.W.: Smart Sensor System and Method Using Surface Acoustic Wave Vapor Sensor Array and Pattern Recognition for Selective Trace Organic Vapor Detection. U.S. Patent 5469369 (1995)
12. Frank, A., Asuncion, A.: UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine, CA, `http://archive.ics.uci.edu/ml/datasets.html`
13. Jha, S.K., Yadava, R.D.S.: Denoising by Singular Value Decomposition and Its Application to Electronic Nose Data Processing. IEEE Sensors J. 11, 35–44 (2011)
14. Ling, C.: Stream data classification using improved fisher discriminate analysis. J. Computers 4, 208–214 (2009)
15. Moradian, M., Baraani, A.: KNNBA: K- Nearest-Neighbor Based-Association Algorithm. J. Theor. Appl. Informat. Technol. 6, 123–129 (2009)
16. Autio, L., Juhola, M., Laurikkala, J.: On the Neural Network Classification of Medical Data and an Endeavor to Balance Non-Uniform Data Sets with Artificial Data Extension. Computers in Biology and Medicine 37, 388–397 (2007)
17. Frank, E., Hall, M.: A Simple Approach to Ordinal Classification. In: De Raedt, L.H., Flach, P. (eds.) ECML 2001. LNCS (LNAI), vol. 2167, pp. 145–156. Springer, Heidelberg (2001)

An Approach Based on Grid-Value for Selection of Parents in Multi-objective Genetic Algorithm

Rahila Patel[1], M.M. Raghuwanshi[2], and L.G. Malik[3]

[1] G.H. Raisoni College of Engineering, Nagpur
rahila.patel@gmail.com
[2] NYSS College of Engineering, Nagpur
m_raghuwanshi@rediffmail.com
[3] G.H. Raisoni College of Engineering, Nagpur, India
lgmalik@rediffmail.com

Abstract. In this work a new approach to parent selection based on Grid-value in multiobjective genetic algorithm is proposed. Here grid is used as a frame to determine the location of individuals in the objective space. Every solution inside the grid maintains an objective-rank vector and summation value. Summation value is the scalar fitness and used to discriminate individuals instead of Pareto-dominance relation. Since multiple solutions occupy same grid have same Summation-value, an adaptive selection mechanism is used in order to avoid duplicate selection and thereby enhancing spread of solution on the Pareto front. The multi-objective genetic algorithm based on the proposed selection scheme is tested on problems of CEC09 competition. The algorithm has shown either comparable or good performance on few unconstrained test problems.

1 Introduction

In multi-objective Genetic algorithm a selection of individual is the crucial issue because the algorithm has to achieve two goals i.e. to maintain diversity along the Pareto optimal front and convergence towards the Pareto optimal front. The selection process in multi-objective GA occurs at two different phases

1. Selection of individuals for reproduction i.e. chosen individuals undergoes crossover and mutation operations. This selection is responsible for convergence towards Pareto-optimal front.
2. Selection of individuals for formation of next generation population. This is concerned with maintenance of diversity among the individuals.

Most of the selection mechanisms used for reproduction and formation of next generation population are based on the fitness of individuals in the population. The fitness of an individual is often how well the individual addresses the objective function of the target problem. Selection usually favors the fitter individuals. [1][2]

Most MOGAs calculate fitness value mainly on the basis of the Pareto dominance relation, i.e., the information which individuals dominates, is dominated or nondominated is used to define a rank [1] [3]. High computational complexity associated with dominance based selection schemes has opened a scope for the development of rank-based

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 265–273, 2011.
© Springer-Verlag Berlin Heidelberg 2011

(non-Pareto) selection schemes, which are less computationally complex. The non-Pareto based methods (VEGA) [4] generate a Pareto set implicitly, without making a direct comparison to check domination/non-domination with other members of the population. There are various non-Pareto based methods called as Rank-based method. Rank-based selection schemes first assign single scalar fitness to each individual and then sort individuals in the population according to single scalar fitness. In [5-11] various rank-based selection have been presented. However, as lack of effective diversity maintenance mechanism, the optimal set obtained by these methods is usually a subset of the Pareto optimal set [15].

Grid has been widely used in the field of evolutionary multiobjective optimization (EMO) due to its property combining convergence and diversity naturally. The recent grid-based EMOAs are presented in [16-19]. In [18] Suganthan *et al* used Summation of Normalized Objective Value for selection of individuals.

Taking idea from [18] and [19] in this work we propose a simple grid-value based selection scheme for Multi-Objective Genetic Algorithm (MOGA). Grid technique is used to balance convergence and diversity in fitness for multi-objective optimization problems. To strengthen selection pressure and refine comparison level, two hierarchical grid-based criterions are incorporated into fitness to establish a complete order among individuals. Moreover, an adaptive selection mechanism in environmental selection is employed to guarantee the diversity of archive memory.

The remainder of the paper is organized as follows. Section 2 briefs about Grid-value based selection scheme. In Section 3, Experimental details are given. Section 4 presents experimental results. Section V draws conclusions.

2 Grid-Value Based Selection Scheme

Grid is an easy way to quantize the search space. Each grid has a grid coordinate. It is a natural way to combine convergence and diversity in evolutionary search since each individual has a determinate location (i.e., coordinates) in it. On one hand, the coordinate not only reflects the number of objectives for which one individual is better than the other, but implies the difference in objective values between the two individuals.

Every solution inside the grid maintains an objective-rank vector and summation value. Summation value based sorting as discrimination technique instead of nondominated sorting. Grid-value based selection is a two phase process:

1. Phase 1 is grid-value calculation
2. Phase 2 is selection based on grid-value

Grid-value calculation is a simple discrimination technique. The idea is to divide every objective's range into 10 grids. All the individuals can have ranks 1 to 10.Then assign objective-rank to individual according to the grid coordinate of the grid to which it belongs. Repeat this for all objectives of solution and calculate sum of all the objective-rank of the solution. This single scalar value is fitness of the solution and we call it as summation-value (fitness) An Algorithm for grid-value calculation is as follows. We assumed that all are minimizing problems (maximizing problems can be converted to minimizing problems by multiplying -1):

- Step 1 Select one unranked objective
- Step 2 Get the maximum and minimum value of the selected objective to calculate the range for this objective
- Step 3 Divide the objective's range into 10 grids.
- Step 4 For every point in the objective space, identify which grid it belongs to and assign the corresponding rank to the point for the selected objective.
- Step 5 If all objectives have been selected go to step 6, otherwise repeat Step 1-4.
- Step 6 Sum the rank of all objectives of each solution to obtain the summation-value of the solution. Obtain summation-value for every solution in the population.

Since multiple individuals are present in the same grid and are having same summation value, a selection strategy is needed which can increase selection pressure towards Pareto front.

An adaptive grid-value based Selection scheme used in the formation of population for next generation. A scan percentage P is set and all the individuals having objective-rank greater than P will not be considered for selection. The proposed adaptive selection scheme adaptively set this scan percentage. In the initial generation P is set to 90 to 80 of total rank and as generation progresses P is gradually reduced to 50 percent of the total rank. The adaptive selection scheme maintains two sets of population namely, preferential set and backup set. All the solutions with rank less than P and minimum Summation-value are placed in the preferential set where as solution with higher Summation-value will be kept in back-up set. Solutions from preference set will be selected as members of next generation. If there is insufficient number of solutions in preference set then solutions from back-up set will be selected as members of next generation. The steps of building the preferential set are as follow:

- Step 1 Select one unselected objective
- Step 2 For the selected objective, scan *P* percentage of the total ranks. For each objective-rank (if this rank is not empty, otherwise just continue to the next rank) of the selected objective, the solutions with the lowest summation-value will be chosen to enter preferential set.
- Step 3 If all objectives have been selected go to step 4. Otherwise repeat Step 1-2.
- Step 4 Collect the solutions not inside the preferential set and put them in the backup set.

Working of the algorithm is illustrated with the sample example as shown in Table 1.

3 Experimentation

An experimental study is performed to investigate the behavior of the proposed approach in terms of convergence and diversity. To demonstrate the proposed scheme of ranking for selection of parents, we have implemented MOGA using MatLab 7.1. The MOGA procedure used for experimentation is given below.

Table 1. Grid-value based selection

S. No	Obj1 rank	Obj2 rank	Sumtion value	Pref set (acco to obj1)	Pref set (accoto obj2)	Preferential set
1	4	3	7	7,2,1,10	10,7,2,3,9,8	1,2,3,7,8,9 and 10
2	3	3	6			
3	3	5	8			
4	9	3	12			
5	3	3	6			
6	3	5	8			
7	1	2	3			
8	4	8	12			
9	9	6	15			
10	8	1	9			

- Step 1. Population (N) initialization.
- Step 2. Evaluate objective functions.
- Step 3. Every individual is assigned scalar fitness by grid-value procedure.
- Step 4. Use Multi-level Tournament selection for selection of individuals for crossover.
- Step 5. Perform crossover in initial generations using MLX (multi-parent crossover with lognormal distribution) and in later generations algorithm uses MPX (multi-parent crossover with polynomial distribution) operator.
- Step 6 Repeat Step 4 and 5 till offspring size equals parent population.
- Step 7. Mix parent and offspring population to produce combined population (2N).
- Step 8. Apply grid-value procedure on combined population.
- Step 9. Select N solutions as parent solution for next generation using adaptive selection method
- Step 10. Check, if the stopping criterion (Maximum No. Of Iterations) is met, then the present parent solutions are Pareto optimal front solutions. Otherwise, the procedure is repeated from Step 2.

Any good search algorithm must explore a large search space in the beginning and the search should then narrow down as it converges to the solution. To support this property explorative operator MLX is used in the initial generations (30-40 percent of maximum generation) and in the later generations exploitative operator MPX is used in the algorithm. Since algorithm is using parent centric multi-parent crossover operators a multilevel tournament selection is used. The steps for multilevel tournament selection are 1)Select two pairs of solutions randomly. 2)Compare summation-value of each pair. The comparison of two pairs will generate two winner solutions. 3)Again compare summation-value of pair winner solutions. This tournament winner is treated as best parent and MLX and MPX [12] generate offspring near this best parent. Multilevel tournament selection increases selection pressure that give better chance to select more fit parents for reproduction. That may help to speed up the convergence.

The test problems (UF1-UF9) in this work are taken from CEC2009 special session and competition [13]. The IGD (Inverted Generational Distance) metric is used as performance indicator to quantify the quality of the obtained results.

PC configuration: Microsoft Windows XP, 2.00 GB RAM and Pentium 4, 2.99GHz

Parameter settings:

- Population size (N): 100 (for 2 objectives) and 150 (for 3 objectives)
- Maximum No of Function Evaluations: 100000
- Crossover probability parameter (p_c): 0.9
- Probability Distribution indices for MPX crossover: 2
- Probability Distribution indices for MLX crossover: 6
- Number of Parents: 5
- Number of children (λ): 2
- Archive size: 200 for two objective problems 300 for three objective problems

4 Disscussion and Comparison of Results

30 random simulations are performed for each problem with grid-value based MOGA The minimum (Min), maximum (Max), mean, and standard deviation (Std) of the IGD metric are reported in Table 2 for function UF1-UF9 with grid-value based MOGA. For seven bi-objective instances, grid-value based MOGA found good approximation to UF2, UF3, UF4 and UF7 but performed poorly on UF5 and UF6. It has given comparable results for UF1. For three objective instances; grid-value based MOGA has shown very poor performance for UF8 and UF9.

Table 2. The IGD Metric for grid-value based MOGA

Problem	Min	Max	Mean	Std
UF1	0.0421287	1.0007224	0.0601896	0.0953393
UF2	0.0203624	0.5467352	0.0170206	0.0164622
UF3	0.0223853	0.0446317	0.0311595	0.0044446
UF4	0.0126238	0.0599791	0.0288905	0.0061184
UF5	0.7748762	3.7762237	0.9789302	0.4070842
UF6	0.2651281	2.8388039	0.3701975	0.2896052
UF7	0.0360071	0.6692672	0.0189312	0.0138001
UF8	1.225456	4.9562607	1.8360914	0.5232838
UF9	0.550592	5.9664864	0.9885287	0.5806802

Grid-value based MOGA algorithm converges to near to Pareto-optimal front with good spread in nondominated solutions on Pareto-optimal front in less than 100000 function evaluations for some functions. Further use of diversified selection of parents for next generation has contributed to good spread of solutions as evident from the IGD values given in the table 2. For other functions to some extent it has found global convergence but the complete Pareto-optimal frontier is not discovered. For functions like UF5 and UF6 Grid-value based MOGA fails to converge to Pareto-optimal front. Both the functions have discontinuous Pareto front. The objective function profile is multimodal near the global Pareto-optimal frontier, and a slight perturbation in the optimization variables causes the solutions to become dominated. Also, the phenomenon of genetic drift causes the population to follow the good solutions, which get discovered early in the search process. This genetic drift results in the clustering of the solutions around these points. Grid-value based MOGA has shown good performance on few bi-objective functions having convex, non-convex and continuous Pareto fronts.

4.1 Comparison Based on Simulation Speed

Further experiments were performed to find computational efficiency of the proposed techniques. For the commonly used non-domination sorting [NDS][3], the complexity to obtain the overall non-dominated set is O (MN^2). For the grid-value method, it requires O(N) comparisons to find the maximum and minimum values and O(N) comparisons to identify the corresponding grid for each objectives. In total, the complexity of each methods is O(MN), where M is the number of objectives and N is the number of solutions. From the complexity calculations, we can observe that the complexities of the grid-value methods are in linear form while non-domination sorting is non-linear. This will reduce the CPU time of the grid-value MOGA. Fig 1 shows the CPU time (in seconds) taken by the grid-value and NDS on problems UF1-UF9.

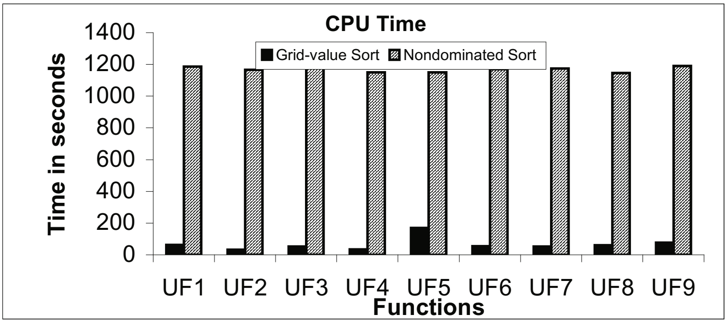


Fig. 1. Comparison of CPU TIME of grid-value and Nondominated sorting techniques

4.2 Rank Distribution

Corne and Knowles [10] proposed to measure the relative entropy of the distribution of ranks induced by a method in order to analyze its effectiveness. It is supposed that a ranking method providing a richer ordering would lead to a better performing optimization scheme. For experimentation population size of 1000 is taken and function used is UF1. Fig 2 shows the relative entropy of the ranking distribution produced by the studied method. Results shown in the fig 2 are average performance of 30 runs.

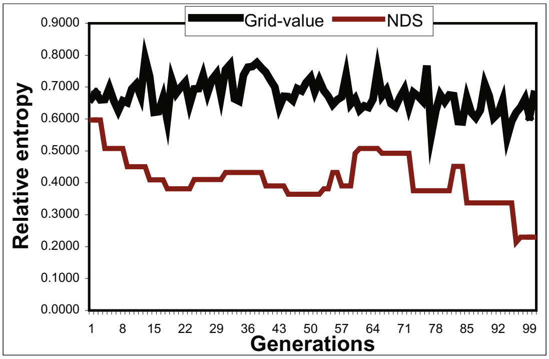


Fig. 2. Comparison of Relative Entropy of Grid-value based MOGA with NDS

From above figure it is clear that grid-value based ranking has better rank distribution than NDS technique. The reason behind this performance is that grid-value technique assigns different ranks to maximum solution in the population and thus grid-value is fine-grained ranking technique than NDS technique.

4.3 Comparison with Other Algorithm

Performance of Grid-value based MOGA is compared with hybrid AMGA[14]. Table 3 shows mean IGD of the algorithms. The statistical test is also performed and shown in the last column of each test function. The numerical values -1, 0, 1 represent that the proposed method is statistically inferior to, equal to and superior to the NDS.

Table 3. Comparison of grid-value MOGA with AMGA and MOEP

Problem	Grid-value based MOGA	AMGA	Statistical test (t-test)
	MeanIGD	MeanIGD	
UF1	0.0601896	0.035886	-1
UF2	**0.0177206**	0.016236	1
UF3	**0.0311595**	0.069981	1
UF4	**0.0288905**	0.040621	1
UF5	0.9789302	**0.094057**	-1
UF6	0.3701975	0.129425	-1
UF7	**0.0189312**	0.057076	1
UF8	1.8360914	**0.171251**	-1
UF9	0.9885287	**0.18861**	-1

Grid-value based MOGA has outperformed the AMGA algorithm on UF2, UF3, UF4 and UF7. For functions UF5 and UF6 Grid-value based MOGA fails, so the primary cause of this behavior is the objective function profile, which is multi-modal near the global Pareto-optimal frontier, and a slight perturbation in the optimization variables causes the solutions to become dominated. Also, the phenomenon of genetic drift causes the population to follow the good solutions, which get discovered early in the search process. This genetic drift results in the clustering of the solutions around these points. From the obvious discussion we can comment that Grid-value based MOGA has performed well on functions having convex, non-convex and continuous Pareto fronts but fails on discontinuous Pareto fronts.

5 Conclusion

In this work we have proposed and implemented a non-Pareto based selection method as an alternative to Pareto-based method. Grid-value based selection is used to discriminate individuals in presence of multiple objectives. Grid-value based MOGA based on above-mentioned selection approach has been tested on unconstrained test problems of CEC09 test suit.

Experimental results indicate that the proposed methods are able to guide the search process towards the optimum for the seven bi-objective and the two 3-objective test functions. Grid-value based MOGA has shown better computational efficiency than non-dominated sort. Grid-value is less complex technique and adaptive selection requires less iteration for selection of individual for next generation.

Grid-value based MOGA has shown better convergence and diversity in most of the test functions having continuous convex and nonconvex Pareto fronts but performed poorly on functions having discontinuous Pareto fronts.

As part of the future work, we want to extend our experiments to investigate and improve the performance of the proposed techniques on the functions having highly multimodal and discontinuous Pareto fronts.

Acknowledgments. We express our special thanks to Dr. P.N.Suganthan, School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, for providing resources, valuable guidance and support in this work.

References

[1] Zhou, A., Qu, B.Y., Li, H., Zhao, S.-Z., Suganthan, P.N., Zhang, Q.: Multiobjective Evolutionary Algorithms: A Survey of the State-of-the-art. Swarm and Evolutionary Computation 1, 32–49 (2011)

[2] Deb, K.: Multi-objective Optimization using Evolutionary Algorithms. John Wiley & Sons, West Sussex (2001)

[3] Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A Fast Elitist Nondominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In: Deb, K., Rudolph, G., Lutton, E., Merelo, J.J., Schoenauer, M., Schwefel, H.-P., Yao, X. (eds.) PPSN 2000. LNCS, vol. 1917, pp. 849–858. Springer, Heidelberg (2000)

[4] Schaffer, J.D.: Multiple objective optimization with vector evaluated genetic algorithms. In: Genetic Algorithms and their Applications: Proc. First Int'l. Conf., pp. 93—100 (1985)

[5] Bentley, P.J., Wakefield, J.P.: Finding acceptable solutions in the Pareto-optimal range using multi-objective geneticalgorithms. In: Chawdry, P.K., et al. (eds.) Soft Computing in Eng'g Design and Manufacturing. Springer, Heidelberg (1997)

[6] di Pierro, F., Soon-Thiam, K., Savic, D.A.: An investigation on preference order ranking scheme for multi-objective evolutionary optimization. IEEE Transactions on Evolutionary Computation 11, 17–45 (2007)

[7] di Pierro, F.: Many-objective evolutionary algorithms and applications to water resources engineering, PhD thesis, University of Exeter, UK (August 2006)

[8] Drechsler, N., Drechsler, R., Becker, B.: Multi-Objective Optimisation Based on Relation Favou. In: Zitzler, E., Deb, K., Thiele, L., Coello Coello, C.A., Corne, D.W. (eds.) EMO 2001. LNCS, vol. 1993, pp. 154–166. Springer, Heidelberg (2001)

[9] Maneeratana, K., Boonlong, K., Chaiyaratana, N.: Compressed-Objective Genetic Algorithm. In: Runarsson, T.P., Beyer, H.-G., Burke, E.K., Merelo-Guervós, J.J., Whitley, L.D., Yao, X. (eds.) PPSN IX. LNCS, vol. 4193, pp. 473–482. Springer, Heidelberg (2006)

[10] Knowles, J., Corne, D.: Techniques for Highly Multi-objective Optimisation: Some Nondominated Points are Better than Others. In: GECCO 2007, London, England, United Kingdom, July 7-11 (2007)

[11] Qu, B.Y., Suganthan, P.N.: Multi-objective Evolutionary Programming without Nondomination Sorting is up to Twenty Times Faster. In: IEEE Congress on Evolutionary Computation-CEC 2009 (2009)

[12] Raghuwanshi, M.M., Kakde, O.G.: Multi-parent Recombination operator with Polynomial or Lognormal Distribution for Real Coded Genetic Algorithm. In: 2nd Indian International Conference on Artificial Intelligence (IICAI), pp. 3274–3290 (2005)

[13] Zhang, Q., Zhou, A., Zhao, S.Z., Suganthan, P.N., Liu, W., Tiwari, S.: Multi-objective optimization Test Instances for the CEC 2009 Special Session and Competition. University of Essex, Colchester, UK and Nanyang Technological University, Singapore, Special Session on Performance Assessment of Multi-Objective Optimization Algorithms, Technical Report (2008)

[14] Tiwari, S., Fadel, G., Koch, P., Deb, K.: Performance Assessment of the Hybrid Archive-based Micro Genetic Algorithm (AMGA) on the CEC 2009 Test Problems. In: Congress on Evolutionary Computation, CEC 2009 (2009)

[15] López Jaimes, A., Coello Coello, C.A.: Study of Preference Relations in Many-Objective Optimization. In: Genetic and Evolutionary Computation Conference (GECCO 2009), pp. 611–618. ACM Press, Montreal (2009)

[16] Zhao, S.Z., Suganthan, P.N.: Two-lbests Based Multi-Objective Particle Swarm Optimizer. Egineering Optimization 43(1), 1–17 (2011)

[17] Li, M., Zheng, J., Shen, R., Li, K., Yuan, Q.: A Grid-Based Fitness Strategy for Evolutionary Many-Objective Optimization. In: GECCO 2010, Portland, Oregon, USA, July 7-11 (2010)

[18] Qu, B.Y., Suganthan, P.N.: Multi-objective Evolutionary Algorithm based on the Summation of Normalized Objectives and Diversified Selection. Information Sciences 180(17), 3170–3181 (2010)

[19] Qu, B.Y., Suganthan, P.N.: Multi-objective Differential Evolution based on the Summation of Normalized Objectives and Improved Diversified Selection. In: The Proc. of Symposium on Difference Evolution, Paris, France (April 2011)

A Novel Non-dominated Sorting Algorithm

Gaurav Verma[1], Arun Kumar[2], and Krishna K. Mishra[3]

[1] NetApp India Pvt Ltd, Bangalore, India
[2] Citrix Systems Pvt Ltd, Bangalore, India
[3] Department of Computer Science and Engineering,
Motilal Nehru National Institute of Technology, Allahabad, 211004, India

Abstract. Many multi-objective evolutionary algorithms (MOEA) require non-dominated sorting of the population. The process of non-dominated sorting is one of the main time consuming parts of MOEA. The performance of MOEA can be improved by designing efficient non-dominated sorting algorithm. The paper proposes Novel Non-dominated Sorting algorithm (NNS). NNS algorithm uses special arrangement of solutions which in turn helps to reduce total number of comparisons among solutions. Experimental analysis and comparison study show that NNS algorithm improves the process of non-dominated sorting for large population size with increasing number of objectives.

1 Introduction

Many modern engineering problems deal with multiple objectives with large solution space. The multi objective optimization process has been proven as a very useful technique in identifying the optimal solutions. Such problems have conflicting objectives which result in a set of optimal solutions, not a single best solution. The set containing these optimal solutions is known as Pareto Front and the solutions in the Pareto Front are called as non-dominated solutions [1]. Over the past 20 years, many multi-objective evolutionary algorithms (MOEAs) were suggested which can be studied in literatures. Many of the multi-objective evolutionary algorithms (NSGA[1], NSGA-II[2], SPEA2[3] and DTEA[4]) require non-dominated sorting to generate non-dominated fronts. This paper explains a novel non-dominated sorting algorithm which is experimentally shown to be time efficient for large population size with increasing number of objectives. In literature, many algorithms show the process for finding non dominated set of solutions such as naïve and slow method [1], Kung et al method [5] and Ding [6]. Recursive application of any of these algorithms would give non-dominated fronts in a sorted order but it would be time consuming as well. In NSGA-II [2], K. Deb et al proposed an algorithm for non-dominated sorting which later in 2005 got improved by Chuan Shi [7]. It's worst case time complexity is $O(MN^2)$ which is the time taken to sort the population into different non-dominated sorted fronts.

In this paper, we propose a completely different algorithm for non-dominated sorting. The paper is organized into four sections. Section 2 presents some background details of non-dominated sorting and specifies the pre-existing algorithms. Section 3 describes the proposed algorithm and explains different steps using a minimization example problem. In Section 4, experimental analysis and complexity of the proposed algorithm are presented. Finally, Section 5 concludes the paper.

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 274–281, 2011.
© Springer-Verlag Berlin Heidelberg 2011

2 Background

2.1 Dominance and Pareto-Optimality

Definition of dominance and pareto optimality have been taken from K. Deb et al book [1]. Here, we define the concept of dominance with the related terms and the definition of non-dominated set:

Definition of Dominance: Solution x1 dominates solution x2, if both the conditions 1 and 2 hold true:

1. Solution x1 is no worse than solution x2 in all objectives, or fj (x1) $\ntriangleright$ fj (x2) for all j $\in$ {1, 2 … M}.
2. Solution x1 is strictly better than solution x2 in at least one objective, or fj (x1) $\triangleleft$ fj (x2) for at least one j $\in$ {1, 2 … M}.

Definition of Non-dominated set: In a solution space *P*, the non-dominated set *Q* contains solutions which are not dominated by any other solution in the space *P*.

2.2 The Fast Non-dominated Sorting Algorithm of NSGA-II (K Deb et al.)

We calculate two entities for each solution in the population. (1) Domination count n_p, the number of solutions which dominate the solution *p*, and (2) S_p, a set of solutions that solution *p* dominates. It needs $O(MN^2)$ comparisons. Fig. 1 describes the fast-non-dominated-sort algorithm of NSGA-II.

```
Fast-non-dominated-sort(P)
  for each $p \in P$
  $S_p = \emptyset$ and $n_p = 0$
  for each $q \in P$
  if ($p$ -< $q$) then                  If $p$ dominates $q$
      $S_p = S_p \cup \{q\}$            Add $q$ to the set of solutions dominated by $p$
  else if ($q$ -< $p$) then
      $n_p = n_p + 1$                   Increment the domination counter of $p$
  if $n_p = 0$ then,                    $p$ belongs to the first front
      $p_{rank} = 1$
      $F_1 = F_1 \cup \{p\}$
  $i = 1$                               Initialize the front counter
  while $F_i \neq \emptyset$
      $Q = \emptyset$                   Used to store the numbers of the next front
      for each $p \in F_i$
            for each $q \in S_p$
                  $n_q = n_q - 1$
                  if $n_q = 0$ then               $q$ belongs to the next front
                        $q_{rank} = i + 1$
                        $Q = Q \cup \{q\}$
                  $i = i + 1$
                  $F_i = Q$
```

Fig. 1.

We will compute its time complexity and also try to look into the places where any improvement can be done. The body of the first inner loop (for each $p \in F_i$) is executed exactly N times as each individual can be the member of at most one front and the second inner loop (for each $q \in S_p$) can be executed at maximum (N - 1) times for each individual which results in overall $O(MN^2)$ computations.

Here, we perform lot of redundant comparisons. We compare each solution $p \in P$ with every other solution $q \in P$ to determine n_p and S_p which would lead us to a situation where redundant comparisons could take place. Suppose solution p has dominated solution q so solution q will be in set S_p. Now if another solution p' dominates solution p then in such case we should not compare solutions p' and q together as solution q would always be dominated by p'. The proposed approach arranges the solutions in such an order which will reduce the number of redundant comparisons significantly.

3 Proposed Algorithm

The proposed algorithm reduces the total number of comparisons significantly by arranging the solutions in an order such that a solution S_j cannot dominate solution S_i, for all $i < j$ in set S containing all elements in the population. It also determines the solutions which will never dominate any other solutions. In order to achieve such a special arrangement of the solutions, we need to follow step 1 and 2 shown in fig. 2 while the rest of the steps give us different non-dominated fronts.

3.1 Novel-non-dominated-sort(P)

```
1. Sort the entire solution space ($P_1 \ldots P_N$) for each objective function ($F_1 \ldots F_M$) and
   create M sorted lists ($O_1 \ldots O_M$).
2. Mark N layers ($L_1 \ldots L_N$) in the above M sorted lists such that layer $L_1$ should hold
   the first solution i.e. best of each sorted list, layer $L_2$ should hold second solution
   i.e. second best of each sorted list and so on.
3. Transfer non-repeated elements from above N layers ($L_1 \ldots L_N$) to a set S of size N
   keeping the order of elements within a layer unchanged. Initialize front values for
   each solution with one and variable end with N.
4. For i = 1 to N-1 do the following steps:
   4.1. If $S_i \in L_{end}$,
            Skip $S_i$ and go to step 4.2
        Else, for each $S_j \in S$ where $i < j \le N$
            If (($S_j$).front == ($S_i$).front) and $S_i$ dominates $S_j$, then
                    ($S_j$).front = ($S_i$).front + 1
   4.2. If all of the elements in $L_{end}$ are skipped/compared in step 4.1, then
            end = end - 1
5. Traverse set S and group elements having same front value to create different non
   dominated fronts.
```

Fig. 2.

While creating the M sorted lists ($O_1 \ldots O_M$), solutions having equal values for the current objective function are arranged according to the next objective function values. If they have different values for the next objective function, we are done, else we have to check the next objective function values and so on. The concept of layering (shown in step 2) can benefit us in determining the solutions which would never dominate any other solution in the remaining population. Any solution residing in the current bottom most layer (called as L_{end}), can't dominate any other solution as it certainly has the worst value for one of the objectives. In step 3, we create a set S of size N with the elements kept in the same order they appeared in the above N layers. This special order will reduce the number of the comparisons. In step 4, for each element S_i, we check whether S_i exists in the current last layer L_{end} or not. We have two possible cases:

1. If S_i exists in L_{end}, it certainly has worst value in any of the objective functions. Using definition 1, we can say S_i can't dominate any other solution. Thus, there is no need of comparing S_i with rest of the elements in the set S.
2. Else, it may dominate other solutions. So we need to compare S_i with the elements having same front values in the set S and update their front values. We will use definition 1 to check the dominance.

```
For each element $S_j \in S$ where $i < j <= N$:
    If (($S_j$).front == ($S_i$).front) and $S_i$ dominates $S_j$, then
        ($S_j$).front = ($S_i$).front + 1
```

In addition to this, we will also update the variable end which gives us the current last layer L_{end}. Layer L_{end} moves from bottom to top direction (i.e. from L_N to L_{N-1} and so on) as soon as all the elements in the current L_{end} layer are done (have their final front value set). Step 5 gives the final sorted non dominated fronts of the given population.

3.2 Minimization Example Problem: Min-Ex

To give a better understanding of the proposed algorithm, we consider the following two objectives, two variable minimization problems in detail and run the proposed algorithm step by step.

Minimize $f_1(x) = x_1$ and Minimize $f_2(x) = (1 + x_2) / x_1$
Subject to $0.1 \le x_1 \le 1$ and $0 \le x_2 \le 5$

Table 1. We have chosen six random solutions in the search space for illustrating the working principle of proposed algorithm

Solution	x_1	x_2	f_1	f_2
P_1	0.31	0.89	0.31	6.10
P_2	0.43	1.92	0.43	6.79
P_3	0.22	0.56	0.22	7.09
P_4	0.59	3.63	0.59	7.85
P_5	0.66	1.41	0.66	3.65
P_6	0.83	2.51	0.83	4.23

Although this problem looks simple, it produces conflicting scenarios between both the objectives, which results in a set of Pareto-Optimal solutions. A straightforward manipulation of the above two functions will allow us to find the following relationship between both objectives:

$$f_2 = (1 + x_2) / f_1$$

Step 1 sorts the above solutions for each objective function f_1 and f_2 which gives us two sorted lists *O*1 and *O*2 corresponding to both the functions respectively. Solutions are arranged in the best to worst order for each function to create the list. Thus, two lists *O*1 = $\{P_3, P_1, P_2, P_4, P_5, P_6\}$ and *O*2 = $\{P_5, P_6, P_1, P_2, P_3, P_4\}$ are created.

L_1: $\{P_3, P_5\}$
L_2: $\{P_1, P_6\}$
L_3: $\{P_2, P_1\}$
L_4: $\{P_4, P_2\}$
L_5: $\{P_5, P_3\}$
L_6: $\{P_6, P_4\}$

Fig. 3.

Step 2 involves creation of layers L_1 to L_N (in this example, N is equal to 6). Here, layer L_1 will contain the first solutions i.e. best from both the sorted lists. Similarly L_2 will contain the second solutions i.e. second best from each list. Finally L_N will contain the worst solutions from each list. The layers from L_1 to L_6 are shown in fig 3.

Step 3 creates a set *S* of length N = 6 by transferring only non-repeated solutions from layer L_1 to layer L_N until set S is full. Here, first element of set S will be the first element of L_1 which is P3. Similarly second element of S will be the next non repeated element in current layer L_1 or subsequent layers (L_2 to L_6) which is P_5 from layer L_1. Now, layer L_1 is exhausted so move to the next layer L_2 and select only non repeated elements until set S gets filled. Finally, Set *S* contains $\{P_3, P_5, P_1, P_6, P_2, P_4\}$. Initially, front value of each element is set to 1 and variable L_{end} is set to last layer L_6.

Step 4 begins with the first element P_3 in set *S* as the current element which does not belong to layer L_{end} (i.e. the last layer L_6) and uses it to dominate elements $\{P_5, P_1, P_6, P_2, P_4\}$ one by one. P_3 only dominates P_4, thus P_4 front value is increased to 2. As P_3 doesn't dominate solutions $\{P_5, P_1, P_6, P_2\}$, their front values will remain unchanged to 1. Different fronts after first iteration are:

Front 1: $\{P_3, P_5, P_1, P_6, P_2\}$.
Front 2: $\{P_4\}$.

In the second iteration, the current element P_5 which does not belong to layer L_{end} (layer L_6), is used to dominate elements $\{P_1, P_6, P_2\}$ one by one as these elements are next to the current element P_5 in the set S and at the same front (equal to one). Fronts:

Front 1: $\{P_3, P_5, P_1, P_2\}$.
Front 2: $\{P_6, P_4\}$.

Similarly in the third iteration, current element P_1 will dominate P_2. Thus front 1 will contain elements P_3, P_5 and P_1 and front 2 will contain elements P_6, P_2 and P_4. In the forth iteration, current element P_6 would get skipped in step 4.1 because it belongs to last layer L_{end}(layer L_6). In the final iteration, current element P_2, which doesn't belong to layer L_{end}, dominates P_4. Different fronts after step 4 are:

Front 1: $\{P_3, P_5, P_1\}$.
Front 2: $\{P_6, P_2\}$.
Front 3: $\{P_4\}$.

Step 5 groups the elements having same front values and creates different non-dominated fronts. We would get following non-dominated fronts:

Front 1: {P_3, P_5, P_1}.
Front 2: {P_6, P_2}.
Front 3: {P_4}.

4 Experimental Results and Complexity Analysis

4.1 Experimental Results

To compare K. Deb et al. algorithm [2] and the proposed novel algorithm, we have used a test problem known as Schaffer's two-objective problem (Schaffer, 1984). We also tried comparing both algorithms along with Chuan Shi [7] by increasing the number of objectives. We used a computer with Intel Dual Core 3.000 GHz CPU and 512MB RAM configuration. To determine the best possible running times for a particular population size and to remove any experimental errors, we took an average of running times of 100 runs for each algorithm.

Schaffer's two-objective problem (Schaffer, 1984) is the most studied single-variable test problem with two objectives. It can be shown as follows:

SCH1: Minimize $f_1(x) = x^2$,
Minimize $f_2(x) = (x-2)^2$, where $-A \leq x \leq A$

Values as low as A = 10 to values as high as $A = 10^5$ have been used in most studies. Here, we have used A = 10. Variable x is generated randomly using random function. Table 2 contains the relationship of running time and population size for the problem SCH1 which is also illustrated in fig. 4. It is clear that the Novel algorithm is faster than K. Deb. et al. algorithm and its running time is much better especially when the population size increases.

With increasing number of objectives and Chuan Shi's Algorithm, novel algorithm is seen much faster. Table 3 shows the running times of these three algorithms with the increasing number of objectives for a population of which size is equal to 1000. Fig. 5 shows the corresponding graph.

Thus, we can see that the Novel algorithm is much faster and more efficient than K. Deb. et al. and Chuan Shi's algorithms, especially in case of large population size and high number objective space. Above experimental data shows clearly that the running time of the Novel algorithm is much better than K. Deb et al. and Chuan Shi's algorithm for both the scenarios, the test problem and increased number of objectives.

Table 2. The relationship of running time (in milliseconds) and population size for test problem SCH1

Population Size	50	100	200	300	400	500	600	700	800	900	1000
Novel Al.	0.077	0.216	0.753	1.640	2.899	4.566	6.698	7.564	9.433	12.91	14.79
K. Deb Et Al.	0.225	1.068	4.376	10.19	15.86	25.17	33.96	45.84	57.69	74.57	92.78

Table 3. The relationship of running time (in milliseconds) with increasing objectives with population size of 1000

M	2	4	6	8	10	12	14	16	18	20
Proposed	13.75	15.91	17.84	21.43	23.6	26.15	27.39	27.85	29.27	29.51
Chuan Shi	40.36	44.1	58.39	75.36	95.0	103.4	115.7	129.8	140.3	158.2
K. Deb	89.5	99.18	125.7	167.9	204.5	210.7	230.1	245.9	290.3	320.5

4.2 Complexity Analysis

Complexity of Novel algorithm is calculated by analyzing the complexity of each step. Step 1 sorts the population in descending order for each objective functions which takes average of $O(MNlog(N))$ running time. Here, we use quick sort algorithm to sort the population for each objective function. Each objective function will take average $O(Nlog(N))$ running time. Thus, M objectives will take $O(M*Nlog(N))$. Step 2 takes $O(MN)$ running time for creating N different layers. Step 3 again takes $O(MN)$ running time because we transfer N unique elements from different layers to set S. Step 4 is the main part of Novel algorithm which assigns different front values to the solutions. This is the most time consuming part of the novel non-dominated sorting algorithm. It is clear that first solution will take max $(M*(N-1))$ comparisons. Similarly, second solution takes max (M*(N-2)) comparisons and so on. The total number of comparisons taken by all the solutions is:

$$= M*(N-1) + M*(N-2) + M*(N-3) \ldots + 1 = M*N*(N-1)/2$$
$$= O(MN^2)$$

Thus, the time complexity of novel algorithm is:

$$= O(MNlog(N)) + O(MN) + O(MN) + O(MN^2)$$
$$= O(MN^2).$$

Above analysis shows that the worst case time complexity of Novel algorithm is equal to $O(MN^2)$ which is true for all existing non-dominated sorting algorithms. But the advantage of this algorithm over others is for the average case scenarios where we reduce the total number of comparisons by arranging solutions in a specific order and identifying the solutions which cannot dominate any other solutions. Hence, the average time complexity of Novel algorithm will always be better than K. Deb's algorithm.

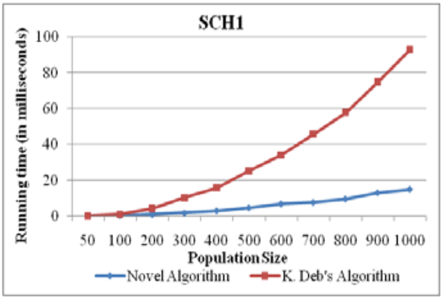


Fig. 4.

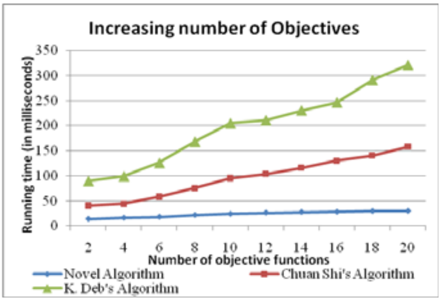


Fig. 5.

5 Conclusion

This paper presents a new non-dominated sorting algorithm, the Novel algorithm. Through the extensive experimental analysis; we assert that the Novel algorithm is better in average running time compared to K. Deb. et al [2] and Chuan Shi's algorithm [7]. When the number of objectives increases, the speedup of the novel algorithm over other algorithms significantly increases. The non dominated fronts generated by the novel algorithm for different numbers of objectives and population sizes were also verified and shown to be identical to those generated by K. Deb. et al.

References

1. Deb, K.: Multi-objective Optimization Using Evolutionary Algorithms, pp. 33–43. John Wiley & Sons, Ltd (2000/2001)
2. Deb, K., Pratab, A., Agarwal, S., Meyarivan, T.: A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)
3. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm. TIK-Report 103. ETH Zentrum, Gloriastrasse 35, CH-8092 Zurich, Switzerland (1999)
4. Shi, C., Li, Y., Kang, L.S.: A New Simple and Highly Efficient multi-objective Optimal Evolutionary Algorithm. In: Proceedings of 2003 IEEE Conference on Evolutionary Computation, Australia (2003)
5. Kung, H., Luccio, F., Preparata, F.: On finding the maxima of a set of vectors. Journal of the Association Computing Machinery 22(4), 469–476 (1975)
6. Freitas, A.A.: A critical review of multi-objective optimization in data mining: a position paper. SIGKDD Explorations 6(2), 77–86 (2004)
7. Shi, C., Chen, M., Shi, Z.: A Fast Non-dominated Sorting Algorithm. In: International Conference on Neural Networks and Brain, ICNN&B 2005, vol. 2, pp. 1605–1610 (2005)
8. Jensen, M.T.: Reducing the run-time complexity of multi-objective EAs: The NSGA-II and other algorithms. IEEE Transactions on Evolutionary Computation 7, 502–515 (2003)
9. 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)
10. Qu, B.-Y., Suganthan, P.N.: Multi-Objective Differential Evolution based on the Summation of Normalized Objectives and Improved Selection Method. In: Proc. of Symposium on Differential Evolution, Paris, France, April 2011

Intelligent Genetic Algorithm for Generation Scheduling Under Deregulated Environment

Sundararajan Dhanalakshmi[1], Subramanian Kannan[1], Subramanian Baskar[2],
and Krishnan Mahadevan[3]

[1] Kalasalingam University, Anand Nagar, Krishnankoil, India
dhana_lakshmi6@yahoo.com, kannaneee@rediffmail.com
[2] Thiagarajar College of Engineering, Madurai, India
sbeee@tce.edu
[3]PSNACollege of Engineering and Technology, Dindugal, India

Abstract. This paper presents an Intelligent Genetic Algorithm (IGA) solution to Generation Scheduling (GS) problem under deregulated environment. In the deregulated market, generating companies (Gencos) will operate with an objective of maximizing their profit, while satisfying the system constraints. Using an intelligent encoding scheme, the minimum up/down time constraints are easily satisfied. Performance of the algorithm is tested on a 10-unit 24-hour unit commitment test system. It is observed from the results, the profit obtained by the proposed algorithm is encouraging to the Gencos.

Keywords: Competitive market, generation dispatch, Genetic algorithm, Optimization.

1 Introduction

In a vertically integrated system, the Generation Scheduling (GS) determines the generating unit's schedules in such a way to minimize the operating costs while satisfying the prevailing constraints such as demand, spinning reserve, minimum up/down time limits and the capacity limits of the generators [1]. Over the years by using several optimization methods, the problem has been solved and there has been savings of huge amount in fuel costs. Deregulation in power sector, increases new investments, improve efficiency, and promotes better pricing in a safe and reliable manner. Hence, many parts of the world are moving towards deregulation, which is the process of separating generation, transmission and distribution from the vertically integrated structure in order to have a better utilization of the resources, introduce choice and to provide quality service to the consumers at economical prices [2],[3]. Hence the main objective of this electricity reform is to introduce competition among various Gencos to enable the generators to compete with each other by availing the open access transmission network [4]. They will also participate in energy contracts directly to supply the generated power to the distributors/bulk consumers with the help of a Power Exchange [5]. Hence, the role of GS is changing and in some aspects it is moving from minimization of cost to maximization of profit [6].

There are two dominant models present in the market; they are Poolco model and Bilateral model [7]. In the Poolco model, all energy and related transmission and ancillary

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 282–289, 2011.
© Springer-Verlag Berlin Heidelberg 2011

services are traded in a central auction mechanism in a coordinated manner and here the Independent System Operator (ISO) is responsible for scheduling the generators [8],[9]. Whereas in a Bilateral model, all energy and related transmission and ancillary services are traded bilaterally, here the GS is done by the individual Gencos. Hence, in this market the GS has a different objective than that of the traditional GS and is referred as Profit Based Generation Scheduling (PBGS) to emphasize the importance of profit.

Different market models proposed under the Poolco/Bilateral market are Consumer Payment minimization, Social Welfare maximization, Price Taker perspective, Stochastic model and Profit maximization model [10-15]. In this paper, PBGS model is considered in view of the fact that most of the researchers used this model. Also, different types of formulations for PBGS problem are available in the literature. K.Bhattacharya et. al proposed a simple model with revenue and cost [9]. Attaviriyanupap et.al used a model which gives more importance to the reserve [11]. Yamin and Shahidehpour proposed a different model, in which the reserve has been split into spinning and non-spinning reserve [12]. In addition, all these models the demand is considered as inequality constraint meaning that there is no obligation to serve the demand. Since, Bhattacharya model is simple and widely accepted [10], [13-15], it is used in this paper.

The traditional method used for solving this problem is Lagrangian Relaxation (LR) similar to the cost based GS [6]. Richter et.al proposed Genetic Algorithm for solving this problem [10]. Later hybrid methods like LR based –Evolutionary Programming, LR based-Genetic Algorithm and LR based-Particle Swarm Optimization [11-13] are used. T.A.A.Victoire et.al used Tabu search based hybrid-optimization technique to solve the PBGS problem [14]. Chandram et.al proposed Improved Pre-prepared Power Demand Table for solving this problem [15]. One of the major difficulties in GS problem is the satisfaction of hard constraints like minimum up and minimum down constraints. In earlier evolutionary algorithms applications to this problem, penalty factor approach is employed for handling these constraints. Most of the computational time is spent for satisfying these constraints. In order to avoid this difficulty intelligent coding scheme is employed for PBGS problem.

Organization of this paper is as follows. Section 2 deals with the problem formulation used for PBGS. Section 3 describes the implementation of IGA to PBGS problem. Section 4 explains the concept of Economic Dispatch used and Section 5 presents the results and discussions and finally Section 6 concludes.

2 Profit Based Generation Scheduling (PBGS) Problem Formulation

The objective of PBGS problem is to maximize the Gencos profit over the scheduled time period, under the generator operational and spinning reserve constraints. The objective function to be maximized is

$$\text{Maximize, } Profit = Revenue - Operating\ \ Cost \tag{1}$$

$$Revenue = \left(f_{pt} \times P_{i,t}\right) U_{i,t} \tag{2}$$

$$Operating\ Cost = \sum_{i=1}^{N}\sum_{t=1}^{T}\left[F_i(P_i^t) + ST_{i,t}(1-U_{i,t-1})\right]U_{i,t} + (1-U_{i,t})SD_{i,t}U_{i,t-1} \tag{3}$$

$$F(P_i) = a_i\,P_i^2 + b_i\,P_i + c_i\ (\$/hr) \tag{4}$$

Subject to,

(i) Inequality Demand constraint

$$\sum_{i=1}^{N} P_{i,t}U_{i,t} \le P_{demand}^{t} \tag{5}$$

(ii) Generation Limit constraint

$$P_{i,\min}\ U_{i,t} \le P_{i,t} \le P_{i,\max}\ U_{i,t} \tag{6}$$

(iii) Spinning Reserve constraint

$$P_{demand}^{t} + R_t = \sum_{i=1}^{N} P_{i,\max}\ U_{i,t} \tag{7}$$

(iv) Minimum Up time

$$U_{i,t} = 1;\ \sum_{j=ts}^{t-1}(1-U_{i.j}) < MUT_i\ ,\ \ for\, i = 1,......N,\ t = t_s + 1,......T \tag{8}$$

(v) Minimum Down time

$$U_{i,t} = 0;\ \sum_{j=td}^{t-1}(1-U_{i,j}) < MDT_i\ ,\ \ for\, i = 1,......N,\ t = t_d + 1,......T \tag{9}$$

3 Intelligent Genetic Algorithm Based PBGS

In general, the application of GA consists of an initialization of population, finding the fitness value of each population, arranging them according to their fitness values, applying genetic operators like reproduction, crossover and mutation to the strings of schedule until the termination criteria is reached [16].

Based on the above discussions the GA is well applied to solve the PBGS problem without constraints. The binary variables $U_{i,t}$ to be determined in the PBGS problem denote the ON-OFF states of the i -th unit at the t-th hour. If the binary decision variables $U_{i,t}$ were directly employed in the GA, the operations of crossover and mutation on the solutions with all constraints satisfied would yield almost all infeasible solutions in the next iteration, particularly for the minimum up and down-time constraints. As a result, the search process becomes very inefficient for the GA to achieve the optimal solution. The binary coded GA with Intelligent Coding is called as Intelligent Genetic Algorithm (IGA) in this paper.

3.1 Encoding Scheme

In the PBGS problem, the decision variables are ON – OFF status of generators, which are represented as binary strings. The constraints present in this problem can be classified into two sets as easy and difficult constraints. The difficult constraints of this problem are (8) and (9). To handle this minimum up and down time constraints effectively, IGA coding scheme is used in this paper. It translates the decision variables U into their binary string representations U_b also in the form of a matrix as shown in Fig. 1. At the same time the minimum up time and down-time are integrated in the representations. Compared with without intelligent coding scheme (direct coding), this intelligent coding scheme, will greatly reduce the searching efforts of the GA [17], [18].

S_{11}		S_{12}		S_{13}		S_{14}			
1	01	0	11	1	00	.	.	.	.

1	101	0	000	1	100	.	.	.	.

0	1001	1	1000	1	0011	.	.	.	.

Fig. 1. The Intelligent Genetic Algorithm coding scheme

As shown in Fig.1, each row of the matrix stands for the coded operating states for one generating unit during the H-hour period. Each row of the matrix is divided into several substrings. e.g., S_{i1}, S_{i2}, …. for the *i*-th row. Each of the substrings indicates one operating state of ON, with a leading bit of 1 or OFF with a leading bit of 0 and the number of hours for which the ON-state or OFF-state lasts besides the minimum up or down time. The length of the substring, n_i for the *i*-th unit is given by,

$$\begin{aligned} n_i &= \lfloor \log_2 (n_{i\max}) \rfloor + 1, \quad && for \quad n_{i\max} > 1 \\ &= 1\,, && otherwise \end{aligned} \tag{10}$$

where $n_{i\max}$ = max (minimum up-time, minimum down-time) for the unit *i*. Hence these minimum up-time and down-time constraints can be included in the binary string of decision variables and are always satisfied implicitly.

For example, unit 1 has a minimum up-time of 2 hours and a minimum down-time of 4 hours. Then, the substring length for the unit is 3 bits. Substring, 1|01, denoted that the unit is in ON state and the up-state lasts for 3 hours, i.e., 1 hour (indicated by the binary value ‘01’) plus 2-hour minimum up-time. Similarly, the substring, 0|11, means that the unit is in OFF-state for 7 hours, i.e., 3 hour (indicated by (‘11’) plus 4-hour minimum down-time. On the other hand, if the substring length is 1, each bit in the corresponding row of the matrix directly represents the hourly schedule of the unit in the H-hour period. After decoding, if the corresponding actual commitment schedule of these units covers a period longer than 24 hours, the scheduling interval beyond the 24th hour is ignored in further evaluation. In addition, the initial status of the generating units (ON or OFF time before the study period), can be included by using the same substrings.

3.2 Implementation of IGA to PBGS Problem

Once these constraints are included, the UC schedules do not have any minimum up/down time constraints and initial status violations. Based on the fitness value, the population strings are chosen for genetic operations such as reproduction, crossover and mutation to create the strings for the next iteration. In this paper, uniform binary crossover is used, in which the bits are randomly copied from the first or from the second parent. It utilizes a crossover template, a random binary vector that is the same length as the parent chromosomes. This template is used to select which genes from each parent should be crossed over. After processing by GA the actual decision variables can be decoded back. This process is repeated till the termination condition is reached.

4 Economic Dispatch

Once the unit commitment status is determined, the economic dispatch is carried out to determine the optimal generation power output for the given time period. The important criteria to be considered in this case of PBGS are that to maximize the profit, it is not necessary to serve the demand [19]. Hence, in order to perform this following procedure is used. First $P_{i,t}$ is calculated using

$$P_{i,t} = \frac{f_{pt}(t) - b(i)}{2a(i)} \qquad for \quad i = 1, 2, \ldots. N \tag{11}$$

If this, $\sum_{i=1}^{N} P_{i,t}\, U_{i,t} \le P_{demand}^{t}$ then P_i's will give dispatch. (12)

$$\text{if, } \sum_{i=1}^{N} P_{i,t} U_{i,t} > P_{demand}^{t} \tag{13}$$

$$\text{then, } \sum_{i=1}^{N} P_{i,t}\, U_{i,t} = P_{demand}^{t} \tag{14}$$

The corresponding $P_{i,t}$ is calculated by using Quadratic Programming method [20]. Similarly, from the decoded schedule the startup cost and shut down cost are arrived. By using the above production cost, start up cost, shut down cost, the total cost and the corresponding revenues are calculated.

5 Simulation Results

The implementation of IGA to the PBGS problem is carried out using MATLAB Version 7.4 on a Pentium dual core processor, desktop computer operating at 2 GHz with 1GB RAM. The effectiveness of the algorithm has been tested on 10 unit test system considered for 24 hours. The maximum number of iterations is set to 200 and P_c, P_m values are set at 0.8 and 0.2 respectively. The detailed fuel cost coefficient, the lower/upper power limits, forecasted market prices and minimum up/down time limits are taken from [6],[11].

Using without intelligent coding, out of ten trials, only six trials give feasible solution with best profit value of 93440 $. Also, the variations in the profits obtained in feasible solutions are very high. Results shows direct coding is not suitable for GS problem. With intelligent coding scheme, the same GA is able to give feasible solutions in all the ten trials. The best and worst profit obtained are 107233$ and 106930$. The variations between the best and worst results are very less as compared to the direct coding. From Table 1, it is observed that, the proposed IGA method provides maximum profits, for the 10-unit 24-hour system when compared to the existing methods. The Table 2 gives the hourly dispatch for 10-unit system with its total revenue, total operating cost and corresponding total profit. From the results, it is clear that the minimum up/down time are satisfied easily by the use of intelligent coding. Also, by using the IGA the minimum up/down time constraints are easily satisfied without any violation, in less computational time.

The convergence characteristic of the 10 unit 24 hour test system with intelligent coding is shown in Fig.2. This figure clearly shows that around 50 iterations GA, with intelligent coding scheme able to find optimum solution. The results obtained using IGA is compared with previous reported results with regard to their profits. In the Table 1, * indicates that results corresponding to the probability of reserve r = 0 is used for comparison.

Table 1. Comparison of Results for 10 unit 24 hour test system

S.No.	Method	Profit ($)
1	TS-RP *[19]	101086
2	TS-IRP * [19]	103261
3	IPPDT[20]	105164
4	IGA	107131

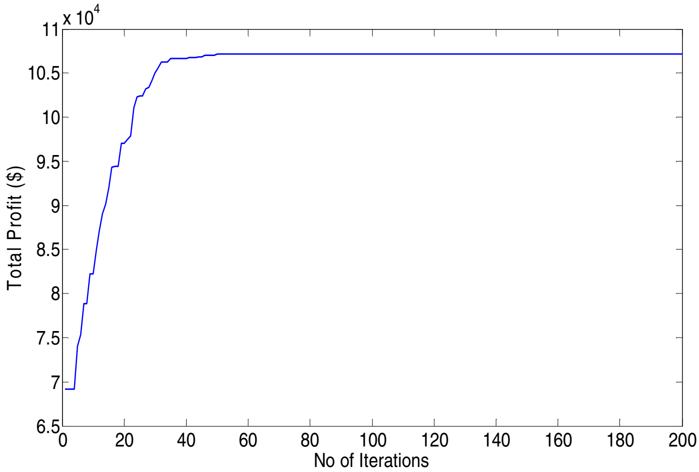


Fig. 2. Convergence characteristics for 10 unit 24 hour test system

Table 2. Hourly Dispatch Results of 10 Unit System

Hour	Generating Unit's Hourly Dispatch (MW)										Total Revenue ($)	Total Cost ($)	Total Profit ($)
	U1	U2	U3	U4	U5	U6	U7	U8	U9	U10			
1	455	245	0	0	0	0	0	0	0	0	**614072**	**506839**	**107233**
2	455	295	0	0	0	0	0	0	0	0			
3	455	395	0	0	0	0	0	0	0	0			
4	455	455	0	0	0	0	0	0	0	0			
5	455	455	0	0	0	0	0	0	0	0			
6	455	455	0	130	0	0	0	0	0	0			
7	455	455	0	130	0	0	0	0	0	0			
8	455	455	0	130	0	0	0	0	0	0			
9	455	455	130	130	130	0	0	0	0	0			
10	455	455	130	130	162	68	0	0	0	0			
11	455	455	130	130	162	80	0	0	0	0			
12	455	455	130	130	162	80	0	0	0	0			
13	455	455	130	130	162	0	0	0	0	0			
14	455	455	130	130	130	0	0	0	0	0			
15	455	455	130	130	0	0	0	0	0	0			
16	455	455	0	130	0	0	0	0	0	0			
17	455	415	0	130	0	0	0	0	0	0			
18	455	455	0	130	0	0	0	0	0	0			
19	455	455	0	130	0	0	0	0	0	0			
20	455	455	0	130	0	0	0	0	0	0			
21	455	455	0	130	0	0	0	0	0	0			
22	455	455	0	130	0	0	0	0	0	0			
23	455	455	0	0	0	0	0	0	0	0			
24	455	345	0	0	0	0	0	0	0	0			

6 Conclusions

Genetic Algorithm with intelligent coding is proposed to Profit Based Generation Scheduling problem with an objective of maximizing the profit of the GENCOs. Intelligent coding scheme is employed to effectively satisfy the minimum up/down time constraints and search efficiency of GA. To demonstrate the effectiveness of IGA, 10 unit 24 hour test system is considered. The results demonstrate that direct coding is not effective for handling minimum up/down constraints of GS problem and necessitates alternate coding strategy. The profit obtained is encouraging and also shows the significance of the proposed algorithm to solve the Generation Scheduling problem in competitive environment.

Acknowledgement. The authors are thankful to the Department of Science and Technology, New Delhi, for its support through the project: SR/ S4/MS/350/06. They are also thankful to the Management of Kalasalingam University and Management of Thiagarajar College of Engineering, Madurai for the facilities provided to carry out this research.

References

[1] Wood, A.J., Wollenberg, B.F.: Power Generation, Operation and Control. John Wiley & Sons (1996)

[2] Yamin, H.Y.: Review on methods of generation scheduling in electric power systems. Elect. Power Syst. Res. 69, 227–248 (2003)

[3] Singh, S.N., Srivastava, S.C.: Electric power industry restructuring in India: present scenario and future prospect. In: IEEE International Conference on Electric Utility Deregulation, Restructuring and Power Technologies (DPRT 2004), vol. 1, pp. 20–23 (2004)

[4] Padhy, N.P., Sood, Y.R.: Advancement in power system Engineering Education and Research with Power Industry moving towards Deregulation. In: IEEE Power Engineering Socitey General Meeting, vol. 1, pp. 71–76 (2004)

[5] Bajpai, P., Singh, S.N.: Electricity Trading In Competitive Power Market: An Overview And Key Issues. In: International Conference on Power Systems, Kathmandu, Nepal, pp. 571–576 (2004)

[6] Shahidehpour, S.M.,Yamin, H.Y., Li, Z.: Market Operations in Electric Power Systems. Wiley (February 2002)

[7] Lai, L.L.: Power system Restructuring and Deregulation. John Wiley and Sons (2001)

[8] Dhanalakshmi, S., Kannan, S., Mahadevan, K.: Market Modes for Deregulated Environment – A Review. In: IEEE Conference ICETECT, Tamilnadu, pp. 82–87 (2011)

[9] Bhattacharya, K., Bollen, M.H.J., Daalder, J.E.: Operation of restructured power systems. Kulwer Academic publisher, Boston (2001)

[10] Richter Jr., C.W., Sheble, G.B.: A profit – Based Unit Commitment Genetic Algorithm for the Competitive Environment. IEEE Trans. Power System 15, 715–721 (2000)

[11] Attaviriyanupap, P., Kita, H., Tanaka, E., Hasegawa, J.: A hybrid LR-EP for solving new Profit Based Unit Commitment Problem under competitive Environment. IEEE Transaction on Power System 18(1), 229–237 (2003)

[12] Yamin, H.Y., Shahidehpour, S.M.: Unit commitment using a hybrid model between Lagrangian relaxation and genetic algorithm in competitive electricity markets. Electric Power Systems Research 68, 83–92 (2004)

[13] Yuan, X., Yuan, Y., Wang, C., Zhang, X.: An improved PSO ap-proach for Profit Based Unit Commitment in Electricity Market. In: IEEE/PES Trans. and Distri. Conference and Exhibition, China (2005)

[14] Victorie, T.A.A., Jeykumar, A.E.: Unit commitment by a tabu-search-based hybrid-optimisation technique. IEE Proc.- Gener. Trans. Distrib. 152(4), 563–574 (2005)

[15] Chandram, K., Subrahmanyan, N., Sydulu, M.: Improved Pre-prepared Power Demand Table and Muller's method to solve the profit based unit commitment problem. Journal of Electrical Engineering & Technology 4(2), 159–167 (2009)

[16] Goldberg, D.E.: Genetic Algorithm in Search, optimization and Machine Learning. Addison-Wesley, MA (1989)

[17] Yang, H.-T., Yang, P.-C., Huang, C.-l.: A parallel Genetic Algorithm Ap-proach in Solving Unit Commitment Problem: Implementation on the Transputer Networks. IEEE Transactions on Power Systems 12, 661–666 (1997)

[18] Baskar, S., Subbaraj, P., Chidambaram, P.: Application of Genetic Algorithms to Unit Commitment problem. IE (I) Journal 81, 195–201 (2001)

[19] Yamin, H.Y.: Profit – based unit commitment for GenCos in a deregulated power market. Euro.Trans. Electr. Power. 14, 321–330 (2004)

[20] Danaraj, R.M.S., Gajendran, F.: Quadratic Programming Solution to Emission and Economic Dispatch Problems. IE (I) Journal 86, 129–132 (2005)

Impact of Double Operators on the Performance of a Genetic Algorithm for Solving the Traveling Salesman Problem

Goran Martinovic and Drazen Bajer

Faculty of Electrical Engineering, Josip Juraj Strossmayer University of Osijek,
Kneza Trpimira 2b, 31000 Osijek, Croatia
`{goran.martinovic,drazen.bajer}@etfos.hr`

Abstract. Genetic algorithms are a frequently used method for search and optimization problem solving. They have been applied very successfully to many NP-hard problems, among which the traveling salesman problem, which is also considered in this paper, is one of the most famous representative ones. A genetic algorithm usually makes use only of single mutation and a single crossover operator. However, three modes for determination which of the double crossover and mutation operators should be used in a given moment are presented. It has also been tested if there is a positive impact on the performance if double genetic operators are used. Experimental analysis conducted on several instances of the symmetric traveling salesman problem showed that it is possible to achieve better results by adaptively adjusting the usage of double operators, rather than by combining any single genetic operators.

Keywords: Combination, genetic algorithm, genetic operators, synergy, traveling salesman problem.

1 Introduction

The traveling salesman problem (TSP) [1] is one of the most famous and most studied combinatorial optimization problems. It has been applied in many fields of science and industry. The TSP consists of finding the shortest tour, visiting each city from a given set of cities only once and returning to the departure city. Although it can be described in a very simple way, the TSP is an NP-hard problem. The TSP also represents a standard benchmark for most algorithmic solutions. For those reasons, many algorithms and improvements of existing algorithms for its solution have been proposed. Since the TSP is NP-hard, solving it to optimality is usually very demanding from the viewpoint of required time and necessary computational resources. This is why approximate methods (heuristics), or more often metaheuristics [2] are used. They cannot guarantee that an optimal solution will be found, but as a rule they find solutions close to optimal ones. One example of successful metaheuristics for the TSP is a genetic algorithm (GA).

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 290–298, 2011.
© Springer-Verlag Berlin Heidelberg 2011

A genetic algorithm is an optimization and search method based on genetics and the process of natural selection, as explained in [3], [4]. GAs have been successfully applied to many problems in different fields of science and industry [3]. Since GAs are inspired by the process of evolution, a population of solutions, in the case of the TSP - a set of traveling salesman tours, advances or evolves throughout generations by applying selection and genetic operators represented by crossover and mutation. Every generation consists of the process of evaluation, selection and recombination. Population solutions, called chromosomes, are evaluated by means of a fitness function that reflects the problem at hand; for the TSP the fitness is usually given by the length of the tour a particular chromosome represents. Selection removes bad solutions from the population, and enables better ones to be subject to recombination through the use of genetic operators. Crossover creates new solutions, i.e. offspring, by exchanging genetic material between two existing ones called parents. Mutation introduces random changes to a chromosome. Its purpose is to keep the population as diverse as possible as well as to recover lost genetic material.

In [5], authors presented an adaptive adjustment mode for the application probabilities of genetic operators. Each operator was first ranked by its performance, and afterwards its application probability was adjusted based on its rank. However, experimental analysis showed that better results could be obtained with static application probabilities. An ensemble of discrete differential evolution (eDDE) algorithms for the generalized TSP was presented in [6]. Each parallel population was assigned a different parameter set for the destruction and construction procedure and a different crossover operator. Analysis showed that better result could be achieved by using eDDE than single populated variants. The application of a GA with multiple crossover and mutation operators on the flow shop scheduling problem was presented in [7]. The usage of each operator was adaptively and dynamically adjusted during the algorithm run. It was shown that in this way better results could be achieved than by using an algorithm with single operators or an algorithm with fixed application probabilities of multiple operators. In [8], authors presented a differential evolution (DE) algorithm in which a pool of mutation and crossover strategies, along with a pool of values corresponding to each associated parameter compete throughout the evolution to produce offspring. Performance was tested on a set of bound-constrained problems, and very promising results were obtained.

This paper shows the impact of using double genetic operators, i.e. double crossover and mutation operators, on the performance of a GA. For the purpose of deciding which of the double genetic operators to use in a given moment, three methods or strategies used to determine the application probabilities of each operator are presented. The performance of those methods was shown on several TSP instances.

Section 2 describes the presented strategies used to determine which of the double genetic operators should be used in a given moment. A TSP definition is given in Section 3. Section 4 deals with encoding of TSP solutions in the GA, used genetic operators, and the mode of operation of the GA implemented for the purpose of performance analysis. Results of the experimental analysis of the impact of double genetic operators on the performance of the GA are given in Section 5.

2 Double Genetic Operators

For the purpose of determining which of the double crossover and mutation operators should be used in a given moment, three methods or strategies for determining their application probabilities are presented. The aforementioned probabilities are not related to algorithm parameter values which affect corresponding operators. For example, this probability only determines which of the two mutation operators has a greater probability to be used, but it does not impact the mutation probability, which presents an algorithm parameter and is equal for both operators. A description of the three presented strategies follows.

Strategy 1. A common method for determining the application probability for one or the other crossover and mutation operator is to set probabilities for both of the operators to equal values. Since double operators are used, the application probability for each crossover and mutation operator is set to 0.5 (50%).

Strategy 2. The application probability of each operator is determined during algorithm execution. Similarly to [5], contribution (in fitness values, i.e. solution qualities) of a particular operator of every three generations is computed. Based on that contribution, the application probability of a particular crossover or mutation operator is increased or decreased (the starting probability for each operator is set to 50%). Increasing the application probability of one crossover or mutation operator (the ones with greater contribution) decreases the application probability of the other crossover or mutation operator. The increase or decrease in the application probability is constant and it means adding or subtracting a value of 0.01 (1%) to or from the current probability value.

The absolute contribution of a particular crossover operator A is given by expression (1), and it represents the average fitness of solutions obtained by this operator. The relative contribution of a particular mutation operator A is given by expression (2), and it represents the average in the difference between solution's fitness after and before it is mutated.

$$x_{avg}^{A} = \frac{\sum_{i=1}^{n} f_o^A(i)}{n}\,. \tag{1}$$

$$m_{avg}^{A} = \frac{\sum_{i=1}^{n} (f_{mo}^{A}(i) - f_o(i))}{n}\,. \tag{2}$$

In expression (1), $f_o^A(i)$ is the fitness of the i-th solution obtained by crossover operator A, and n is the total number of solutions in the last three generations obtained by this operator. In (2), $f_{mo}^{A}(i)$ denotes the fitness of the i-th solution after mutation operator A has been applied on it, whereas $f_o(i)$ is the fitness of the same solution before mutation took place, and n is the total number of solutions in the last three generations on which mutation operator A was applied.

Strategy 3. As in strategy 2, the application probability of each operator is adjusted during algorithm execution. However, the increase or decrease of the application probability of a particular crossover operator is not constant, but it is determined according to expression (3). The increase or decrease of the application probability of a particular mutation operator is not constant either, but it is determined according to expression (4). The increase or decrease of the application probability of each operator given by expressions (3) and (4) is based on the contribution of those operators, i.e. expressions (1) and (2).

$$\Delta p_A = \frac{x_{avg}^B - x_{avg}^A}{x_{avg}^A + x_{avg}^B} \, . \tag{3}$$

$$\Delta p_A = -\frac{m_{avg}^B - m_{avg}^A}{|m_{avg}^A| + |m_{avg}^B|} \, . \tag{4}$$

In expressions (3) and (4), Δp_A is the amount of application probability increase or decrease for crossover operator A or mutation operator A. Increasing or decreasing the application probability of operator A, simultaneously decreases or increases the application probability of operator B.

Strategy 1 does not favor any operators, while strategies 2 and 3, similarly to [5], [7], favor better crossover and mutation operators, i.e. operators that produce higher quality solutions. The total application probabilities for crossover, i.e. mutation operators, are 1. Also, the maximal application probability of each operator in strategy 2 and 3 is 0.9 (90%), which implies that the minimal is 0.1 (10%), the reason being that operators that eventually contributed less to the overall population fitness should not be completely neglected.

3 Traveling Salesman Problem

Formally, the TSP may be described as a permutation problem. If labels $\{1,2,\dots,n\}$ of n cities that are to be visited and matrix $D = [d_{i,j}]$ of dimensions $n \times n$ are given, where $d_{i,j}$ represents the distance between cities i and j, and where $i,j = 1,2,\dots,n$, then the TSP means finding the permutation (detour sequence) σ such that expression (5) is minimal.

$$\sum_{i=1}^{n-1} d_{\sigma(i),\sigma(i+1)} + d_{\sigma(n),\sigma(1)} \, . \tag{5}$$

If the distance between at least one pair of cities i and j is not the same in both directions ($d_{i,j} \neq d_{j,i}$), then we refer to the asymmetric TSP (ATSP); otherwise we refer to the symmetric TSP (STSP). In this paper, we consider the Euclidean TSP as an example of the STSP, where the cities represent points in $\mathbb{R}^2$ (more generally in $\mathbb{R}^d$), i.e. in the Euclidean plane. Distances between cities are represented as Euclidean distances between corresponding points in the plane.

4 Experiment Setup

For the purpose of showing the impact of using double genetic operators, a hybrid GA [9] was implemented in the C# programming language. Since the TSP is considered as a permutation problem, corresponding genetic operators were used. The following operators were used, whose modes of operation are described in [3], [10]: partially matched crossover (PMX), order crossover (OX), insertion mutation (ISM) and exchange mutation (EM). Based on the aforementioned, permutation encoding was used for the representation of TSP solutions in the GA, where every chromosome represents a detour sequence, i.e. a string of labels of cities to be visited in that order. According to [3], [10], PMX respects the absolute position, whereas OX respects the relative position of city labels in the string or detour sequence. EM exchanges two randomly chosen city labels in the string, whereas ISM moves a randomly chosen city label to a new, also randomly chosen, position in the string.

4.1 Genetic Algorithm Used

The mode of operation of the used hybrid GA is shown in Figure 1. A short description of used algorithm follows.

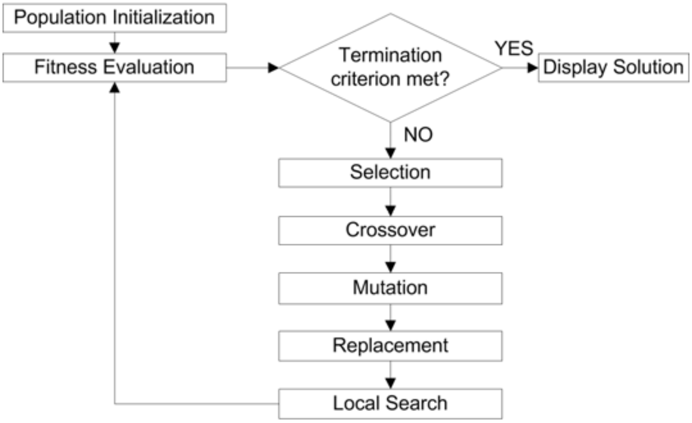


Fig. 1. Hybrid genetic algorithm

Population Initialization. The initial population of size N is filled with randomly created solutions, whereby no duplicates are created.

Fitness Evaluation. The fitness of every solution, i.e. chromosome, is given by a negative value of the length of the tour it represents.

Termination Criterion. Algorithm execution is terminated after a set number of iterations (generations) n_G.

Selection. Solutions to be removed from the population are chosen by a 3-tournament elimination selection without duplicates, as described in [11]. Since the best solution cannot be removed from the population by this selection procedure, elitism is inherent, i.e. the currently best solution in the population is always kept.

Crossover. Before crossover takes place, it is checked if both chosen parents have the same fitness values (if they have the same fitness values, they probably represent the same solutions). If they do not have the same fitness values, they are crossed over; otherwise one parent is mutated before crossover takes place. Although the used crossover operators create two new solutions, only the better one survives.

Mutation. Mutation is applied with a set mutation probability, p_m. Before mutation takes place, it is checked if the chromosome, i.e. solution, acquired by crossover is better than the currently best in the population. In case it is better, it is not mutated (this is simpler and faster than to apply mutation and reject the mutant if it does not improve on the currently best solution); otherwise it is mutated, but only if none of the parents were mutated before.

Replacement. The solution chosen for removal, i.e. the worst solution participating in the 3-tournament selection, is replaced by an offspring of the other two participants. This kind of selection enables selection and recombination to take place in the same step, as described in [11]. Mortality M determines the percentage of population to be replaced by new solutions.

Local Search. Every solution in the set that makes the population is improved by a 2-opt algorithm [12]. The 2-opt algorithm is applied after every two generations.

5 Experimental Results

Experimental analysis was conducted on a computer with an Intel T9400 (2x2.53 GHz) processor, 4 GB of RAM and a 64-bit version of Windows Vista. Only one processor core was used during algorithm execution due to sequential algorithm implementation. The analysis of algorithm efficiency was done on several different types of Euclidean TSP instances taken over from [13].

Parameter values of the GA used during the experimental analysis and acquired by a preliminary analysis are given in Table 1. Experimental results of the conducted analysis are shown in Table 2. They were obtained based on 30 algorithm runs for every combination operators/strategy for a corresponding TSP instance. Table 2 gives TSP instances (the numeric value in the name represents the problem size), their optimal solutions (OPT), as well as the GA setting, i.e. whether single operators are used and which ones (e.g. OX & EM) or whether double operators are used and which strategy (an instance of a algorithm proposed by strategy 1-3 looks like: (OX or PMX) & (EM or ISM) in relation to instances with single operators. Also, Table 2 gives qualities (lengths) of the best obtained solutions (L_{bs}), qualities of average solutions (L_{avg}), qualities of the worst obtained solutions, as well as the deviation of average solutions from the optimal ones (V) and average algorithm execution times (t). The best average solution for each problem instance is given bold.

Table 1. Parameter values of the used GA

Parameter	n_G	N	M	p_m
Value	500	30	0.5 ($N/2$)	0.03 (3%)

According to Table 2 and Table 3, it can be seen that for the majority of problem instances best average solutions were obtained by using strategy 3. Exceptions represent instances kroA200 and a280. Better results were achieved by using strategy 3 than strategy 2, which means that it is better to adjust application probabilities of particular operators based on their contribution, than to do so in constant steps. Also, as can be seen in Table 4, the average execution time for strategies 2 and 3 throughout

Table 2. Experimental results

Instance	OPT	Operators/strategy	L_{bs}	L_{avg}	L_{ws}	$V[\%]$	t[s]
kroA150	26524	OX & ISM	26524	26563.80	26795	0.15	8.02
		OX & EM	26524	26574.93	26682	0.19	6.52
		PMX & ISM	26524	26565.70	26756	0.16	8.22
		PMX & EM	26524	26675.97	26966	0.57	5.03
		Strategy 1	26524	26552.60	26675	0.11	7.88
		Strategy 2	26524	26556.40	26682	0.12	8.37
		Strategy 3	26524	**26548.70**	26655	0.09	7.97
kroA200	29368	OX & ISM	29368	29461.17	29717	0.32	16.12
		OX & EM	29368	29490.60	29680	0.42	13.19
		PMX & ISM	29368	29443.50	29719	0.26	16.07
		PMX & EM	29394	29659.17	30274	0.99	9.65
		Strategy 1	29368	**29441.27**	29618	0.25	15.49
		Strategy 2	29368	29470.40	29776	0.35	15.93
		Strategy 3	29368	29.466.27	29782	0.33	15.17
pr226	80369	OX & ISM	80369	80379.20	80643	0.013	19.91
		OX & EM	80369	80411.03	80645	0.052	15.98
		PMX & ISM	80369	80400.47	80729	0.039	21.06
		PMX & EM	80377	80717.40	81291	0.43	12.37
		Strategy 1	80369	80388.87	80608	0.025	20.75
		Strategy 2	80369	80395.33	80608	0.033	20.24
		Strategy 3	80369	**80378.17**	80608	0.011	20.28
a280	2579	OX & ISM	2579	2591.27	2636	0.48	30.28
		OX & EM	2579	2589.27	2630	0.40	25.94
		PMX & ISM	2579	**2586.57**	2605	0.29	33.24
		PMX & EM	2587	2615.70	2681	1.42	21.24
		Strategy 1	2579	2590.27	2611	0.44	31.08
		Strategy 2	2579	2590.07	2616	0.43	31.29
		Strategy 3	2579	2588.73	2611	0.38	30.86
rd400	15281	OX & ISM	15321	15447.30	15588	1.09	78.20
		OX & EM	15357	15471.03	15591	1.24	64.35
		PMX & ISM	15361	15446.20	15579	1.08	90.24
		PMX & EM	15518	15667.00	15974	2.53	58.85
		Strategy 1	15322	15445.27	15556	1.07	81.60
		Strategy 2	15316	15443.70	15592	1.06	86.43
		Strategy 3	15368	**15424.67**	15531	0.94	81.55
pcb442	50778	OX & ISM	50983	51275.17	51682	0.98	98.38
		OX & EM	50965	51366.27	52014	1.16	81.12
		PMX & ISM	51018	51279.27	51722	0.99	112.44
		PMX & EM	51338	52044.20	52772	2.49	74.36
		Strategy 1	50998	51219.50	51566	0.87	102.17
		Strategy 2	51053	51264.50	51836	0.96	103.67
		Strategy 3	50948	**51218.77**	51522	0.87	105.71
rat575	6773	OX & ISM	6849	6903.63	6955	1.93	212.02
		OX & EM	6854	6919.90	6962	2.12	172.70
		PMX & ISM	6854	6904.23	6949	1.94	258.74
		PMX & EM	6945	7067.13	7138	4.34	170.92
		Strategy 1	6835	6902.27	6977	1.91	218.85
		Strategy 2	6852	6897.90	6946	1.84	221.34
		Strategy 3	6852	**6897.07**	6941	1.83	231.70

all problem instances is approximately the same, while strategy 1 has a minor advantage concerning that point. Adjusting application probabilities for particular operators based on their contribution enables a fast increase or decrease in the aforementioned probability for particular crossover and mutation operators if one operator was superior to another. It also enables a meaningless change in the application probability if both crossover or mutation operators contributed to the population in the same amount. Surprisingly good results were achieved by using strategy 1, where application probabilities of all operators were equal and constant in time. Thereby, this strategy is also the easiest to implement.

A positive impact of using double crossover and mutation operators, i.e. their synergy, is best observed on bigger problem instances, such as rd400, pcb442 and rat575, where all of the described strategies achieved better results than any combination of single operators. Best average results using single crossover and mutation operators were achieved by operators PMX & ISM, which according to Table 4, had the longest average execution time, whereas by far the worst results were obtained by using operators PMX & EM, which however had the shortest average execution time.

According to the experimental results obtained, it was shown that it is possible to achieve high-quality solutions to the TSP by using a hybrid GA with an adaptive adjustment of double genetic operators' usage. From the aforementioned it can be concluded that synergy was achieved between different crossover and mutation operators, which explore the solution space in different manners.

Table 3. Strategies for determining application probabilities of operators and operator combinations ranked by performance

Operators / Strategy	Average deviation from optimal solutions throughout all problem instances [%]
Strategy 3	0.64
Strategy 1	0.67
PMX & ISM	0.68
Strategy 2	0.69
OX & ISM	0.70
OX & EM	0.80
PMX & EM	1.83

Table 4. Strategies for determining application probabilities of operators and operator combinations ranked by average execution time

Operators / Strategy	Average execution time throughout all problem instances [s]
PMX & EM	50.35
OX & EM	54.26
OX & ISM	66.13
Strategy 1	68.26
Strategy 2	69.61
Strategy 3	70.46
PMX & ISM	77.15

6 Conclusion

A GA with double genetic operators for solving the TSP was presented in this paper. Three strategies for determining application probabilities of each used crossover and mutation operator were given. In the first strategy, the application probability of each genetic operator was equal and constant in time. The other two strategies adjusted application probabilities adaptively during the algorithm run, and favored better operators. Experimental analysis, conducted on several TSP instances, showed that higher quality solutions were obtained by using an adaptive adjustment of application probabilities of double operators, rather than by using any combination of single operators. In future research we plan to examine if higher or lower synergy is achieved by using some other crossover and mutation operators.

Acknowledgments. This work was supported by research project grant No. 165-0362980-2002 from the Ministry of Science, Education and Sports of the Republic of Croatia.

References

1. Gutin, G., Punnen, A.P.: The Traveling Salesman Problem and Its Variations. Kluwer Academic Publishers, New York (2004)
2. Metaheuristics Network, `http://www.metaheuristics.net/index.php?main=1`
3. Sivanandam, S.N., Deepa, S.N.: Introduction to Genetic Algorithms. Springer, Heidelberg (2008)
4. Haupt, R.L., Haupt, S.E.: Practical Genetic Algorithms, 2nd edn. John Wiley and Sons, New Jersey (2004)
5. Tuson, A., Ross, P.: Cost Based Operator Rate Adaption: An Investigation. In: Ebeling, W., Rechenberg, I., Voigt, H.-M., Schwefel, H.-P. (eds.) PPSN 1996. LNCS, vol. 1141, pp. 461–469. Springer, Heidelberg (1996)
6. Tasgetiren, M.F., Suganthan, P.N., Pan, Q.K.: An Ensemble of Discrete Differential Evolution Algorithms for Solving the Generalized Traveling Salesman Problem. Applied Mathematics and Computation 215(9), 3356–3368 (2010)
7. Zhang, L., Wang, L., Zheng, D.-Z.: An Adaptive Genetic Algorithm with Multiple Operators for Flow Shop Scheduling. Int. J. of Advanced Manufacturing Technology 27(5-6), 580–587 (2006)
8. Mallipeddi, R., Suganthan, P.N., Pan, Q.K., Tasgetiren, M.F.: Differential Evolution Algorithm with Ensemble of Parameters and Mutation Strategies. Applied Soft Computing 11(2), 1679–1696 (2011)
9. El-Mihoub, T.A., Hopgood, A.A., Nolle, L., Battersby, A.: Hybrid Genetic Algorithms: A Review. Engineering Letters 13(2) (2006), `http://www.engineeringletters.com/issues_v13/issue_2/EL_13_2_11.pdf`
10. Larranaga, P., Kuijpers, C.M.H., Murga, R.H., Inza, I., Dizdarevic, S.: Genetic Algorithms for the Traveling Salesman Problem: A Review of Representations and Operators. Artificial Intelligence Review 13, 129–170 (1999)
11. Golub, M., Jakobovic, D., Budin, L.: Parallelization of Elimination Tournament Selection without Synchronization. In: Proc. of the 5th IEEE Int. Conf. on Intelligent Engineering Systems, INES 2001, Helsinki, Finland, September 16-18, pp. 85–89 (2001)
12. Johnson, D.S., McGeoch, L.A.: The Traveling Salesman Problem: A Case Study in Local Optimization. In: Aarts, E.H.L., Lenstra, J.K. (eds.) Local Search in Combinatorial Optimization, pp. 215–310. John Wiley and Sons, London (1997)
13. Ruprecht-Karls-Universität Heidelberg, `http://www2.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/`

Parent to Mean-Centric Self-Adaptation in SBX Operator for Real-Parameter Optimization

Himanshu Jain and Kalyanmoy Deb

Department of Mechanical Engineering,
Indian Institute of Technology Kanpur,
Kanpur, PIN 208016

Abstract. Most real-parameter genetic algorithms (RGAs) use a blending of participating parent solutions to create offspring solutions in its recombination operator. The blending operation creates solutions either around one of the parent solutions (having a parent-centric approach) or around the centroid of the parent solutions (having a mean-centric approach). In this paper, we argue that a self-adaptive approach in which a parent or a mean-centric approach is adopted based on population statistics is a better procedure than either approach alone. We propose a self-adaptive simulated binary crossover (SA-SBX) approach for this purpose. On a test suite of six unimodal and multi-modal test problems, we demonstrate that a RGA with SA-SBX approach performs consistently better in locating the global optimum solution than RGA with original SBX operator and RGA with mean-centric SBX operator.

Keywords: Self Adaptation, Real coded Genetic Algorithms, Crossover.

1 Introduction

One of the challenges in designing an efficient real-parameter evolutionary optimization algorithm is the recombination operator in which two or more population members are blended to create one or more new (offspring) solutions. The recombination operators can be classified into two functionally different classes based on the location of creating offspring solution vis-a-vis the location of parent solutions. An earlier study [2] classified the real-parameter recombination operators into two classes: (i) mean-centric operators and (ii) parent-centric operators. In a mean-centric operator, offspring solutions are created around the mean of the participating parent solutions (variable or vector-wise). BLX [4], UNDX [7] and SPX [6] operators can be considered to be mean-centric recombination operators. In a parent-centric operator, offspring solutions are created around one of the parent solution (variable or vector-wise). FR [8], SBX [1], PCX [2] are examples of such an operator. Since parent solutions are tested to be better by the preceding selection operator, performing a parent-centric operator was argued to be a better operator than a mean-centric operator. This is particularly true in the beginning of a simulation when parent solutions are away from each other and the mean of parent solutions is expected to lie completely

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 299–306, 2011.
© Springer-Verlag Berlin Heidelberg 2011

in a new region which was not adequately tested for its superiority by any of the parent solutions.

In this paper, we argue and demonstrate that while a parent-centric operator may be judged to be better early on in a simulation, a mean-centric operator may be found to be beneficial when population members crowd around the optimum. Thus, instead of using a parent-centric or a mean-centric recombination operator throughout a simulation, a better strategy would be to adaptively choose a parent or a mean-centric operator depending on whether the population is approaching or is crowded around the optimum.

1.1 Simulated Binary Crossover (SBX)

SBX is a variable-wise recombination operator and was proposed by Deb et al. in 1995 [1].In this operator for two parent values of a particular variable, solutions close to each parent are chosen with a polynomial probability distribution, as shown in Figure 1. This operator has the ergodic property such that any real value in the search space can be created from any two parent values, but with differing probabilities. The parameter η controls the diversity in the offspring values compared to that in the parent values.

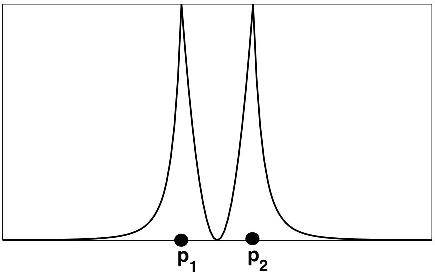


Fig. 1. SBX-η) operator. η is a user-defined parameter

Since we have modified this operator in our study here, we describe this operator in somewhat more details. For two participating parent values (p_1 and p_2) of the i-th variable, two offspring values (c_1 and c_2) are created as a linear combination of parent values, as follows:

$$c_1 = 0.5(1 - \beta(u))p_1 + 0.5(1 + \beta(u))p_2, \tag{1}$$

$$c_2 = 0.5(1 + \beta(u))p_1 + 0.5(1 - \beta(u))p_2. \tag{2}$$

The parameter $\beta(u)$ depends on a random number u created within [0,1] for each recombination operation, and is given below:

$$\beta(u) = \begin{cases} (2u)^{\frac{1}{\eta+1}}, & \text{if } u \leq 0.5, \\ \left(\frac{1}{2(1-u)}\right)^{\frac{1}{\eta+1}}, & \text{otherwise}. \end{cases} \tag{3}$$

The resulting weights to p_1 and p_2 are biased in such a way that values close to the parent values are more likely than values away from them, thereby providing SBX its parent-centric property.

2 Parent or Mean-Centric Recombination

Although both types of recombination operators exist, it is important to understand under what scenarios each type of recombination operator will be useful. Let us investigate two scenarios for this purpose. Figure 2 shows that an EA population is approaching the optimum. In such a scenario, the population-best solution is likely to lie at the boundary of the current population, particularly when the population is lies within an optimal basin. Ideally, in such a scenario, a parent-centric recombination is likely to yield better offspring solutions, as solutions beyond the population-best solution and closer to the optimum are likely to have better objective value. Thus, while approaching an optimum, it is better to employ a parent-centric recombination operator.

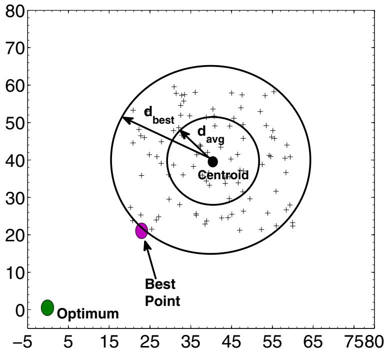


Fig. 2. Scenario 1: Population is approaching the optimum

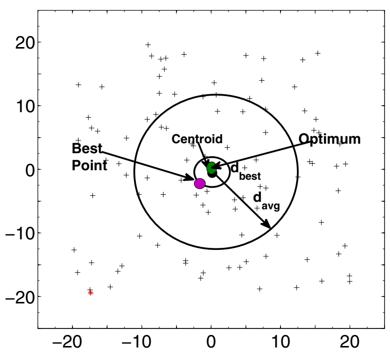


Fig. 3. Scenario 2: Population surrounds the optimum

On the other hand, Figure 3 shows a scenario in which the EA population surrounds the optimum and the population best is likely to be an intermediate population member. In such a case, a mean-centric recombination operator is likely to produce better solutions. Thus, while surrounding an optimum, it is better to employ a mean-centric recombination operator.

While the above two scenarios and corresponding nature of recombination operations are somewhat clear, it is not clear how to know which scenario exist at any particular generation so that a suitable type of recombination operator can be applied. In this study, we compute a parameter λ for this purpose based on the location of population-best and population-average points at every generation.

$$\lambda = \frac{d_{\text{best}}}{d_{\text{avg}}}, \tag{4}$$

where d_{best} is distance of population-best solution from the centroid of population in the variable space and d_{avg} is the average distance of population members from the centroid of population. A little thought and both figures will reveal that in the first scenario, λ is likely to be greater than one and in the second scenario it is likely to be smaller than one. Since the parameter λ captures the extent of approach to or crowding near the optimum by the current population, we use this parameter to redefine creation of offspring in the SBX operator.

3 Self-Adaptive SBX Operator

We consider the SBX operator and modify it using the λ value. First, we create two virtual parents v_1 and v_2 from two participating parents p_1 and p_2 as follows:

$$v_1 = \frac{p_1 + p_2}{2} - \lambda \frac{p_2 - p_1}{2}, \tag{5}$$

$$v_2 = \frac{p_1 + p_2}{2} + \lambda \frac{p_2 - p_1}{2}. \tag{6}$$

The virtual parents (instead of p_1 and p_2) are then used to create two offspring values using equations 1 and 2. Unlike in another study [3], the parameter η of the SBX parameter is kept the same from the start to the end of a simulation.

Thus, if $\lambda = 0$ (population-best is at the centroid of the population), two virtual parents are identical and at the mean value of the two parent values. This causes a mean-centric operation. On the other hand, if $\lambda = 1$ (population-best is at the average distance from centroid of the population), virtual parents are identical to the original parents. This makes the adaptive SBX to be identical to the original SBX operator. For $\lambda > 1$ (population-best lying around the edge of the population), virtual parents move outside the original parent values and the created offspring values are likely to have a greater diversity than the original SBX operation.

Since the parameter λ is not pre-fixed and depends on the location of population-best solution from the centroid of the population and a measure of an average diversity of the population, λ value will change from one generation

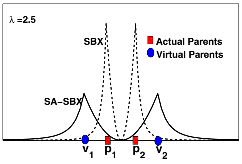


Fig. 4. Distribution of offspring values for Scenario 1

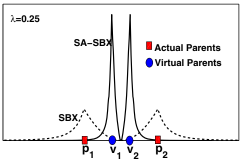


Fig. 5. Distribution of offspring values for Scenario 2

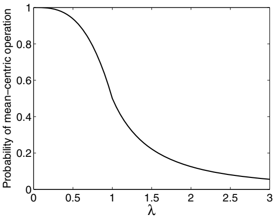


Fig. 6. Probability of mean-centric operation as a function of λ

to another. If the population-best solution is at the periphery of the population (as in scenario 1 in Figure 2), the adaptive SBX creates a diverse set of offspring solutions and the search tends to explore the search space beyond the region represented by the population. Figure 4 shows the distribution of offspring values for a particular variable. However, when the population-best solution is close to the centroid of the population at any generation (as in scenario 2 in Figure 3), the above adaptive SBX behaves like a mean-centric operator and diversity of the created offsprings is expected to reduce, thereby focusing the search. Figure 5 shows the corresponding distribution of offsprings. If we plot the probability of creating offspring between parents (mean-centric) as a function of λ (see Figure 6) we can clearly see that for $\lambda < 1$ probability is greater than 0.5 (i.e mean-centric) and for $\lambda > 1$ probability is smaller than 0.5 (i.e parent-centric).

Interestingly, all these happen adaptively and without any user intervention. This property makes the modified SBX a self-adaptive operator. We now present simulation results using the adaptive-SBX operator.

4 Results

To check the performance of proposed adaptive SBX we choose four unimodal and two multi-modal test problems from the EA literature. They are tabulated in Table 1. Further two variations of the recombination operator: RGA with the original SBX operator (a purely parent-centric operator) and RGA with a purely mean-centric operator are used so as to compare the performance of the proposed adaptive SBX. In each case, 20 runs are made from different initial populations until a specified number of function evaluations are elapsed. Thereafter, best, median and worst objective value attained in 20 runs are tabulated and plotted. In all cases, a population of size $5n$ (where n is the number of variables in the problem), a crossover probability of 0.9, and $\eta = 2$ are followed. To evaluate the effect of recombination operator alone, we have not used any mutation operator in this study. For all test problems, we have chosen two initialization processes: (i) Symmetric initialization around optimum: $x_i \in [-20, 20]$ for all variables, and (ii) One-sided initialization away from optimum: $x_i \in [20, 60]$ for all variables. Each problem is considered with different sizes, n = 20, 50, and 100.

Table 1. Unimodal and Multi-modal test problems used in the study

Prob.	Objective function	f^*
	Unimodal Problems	
P1	$f_{\text{ellipsoidal}} = \sum_{i=1}^{n} i x_i^2$	0
P2	$f_{\text{discus}} = 10^4 x_1^2 + \sum_{i=2}^{n} x_i^2$	0
P3	$f_{\text{cigar}} = x_1^2 + 10^4 \sum_{i=2}^{n} x_i^2$	0
P4	$f_{\text{ridge}} = x_1^2 + 100\sqrt{\sum_{i=2}^{n} x_i^2}$	0
	Multi-modal Problems	
P5	$f_{\text{griewangk}} = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}} + 1\right)$	0
P6	$f_{\text{schaffer}} = \left(\frac{1}{n-1}\sum_{i=1}^{n-1} \sqrt{s_i} + \sqrt{s_i}\sin^2(50 s_i^{1/5})\right)^2, \quad s_i = \sqrt{x_i^2 + x_{i+1}^2}$	0

Table 2. Objective function value (median of 20 runs and best & worst in brackets) for all three) for all three algorithms are shown for problems with symmetric initialization

Function	Initialization	n	Function Evaluations	SA-SBX	parent-centric SBX	mean-centric SBX
$P1$	$[-20, 20]^n$	20	20,500	$\mathbf{1.90 \times 10^{-8}}$	5.59×10^{-6}	2.91
		50	87,500	$\mathbf{1.20 \times 10^{-8}}$	4.61×10^{-2}	2.13×10^{1}
		100	290,000	$\mathbf{1.10 \times 10^{-8}}$	5.67×10^{3}	1.49×10^{2}
	$[20, 60]^n$	20	23,000	$\mathbf{1.89 \times 10^{-6}}$	2.51×10^{-5}	4.45×10^{4}
		50	111,250	$\mathbf{1.30 \times 10^{-8}}$	1.04×10^{-2}	3.39×10^{5}
		100	370,000	$\mathbf{1.30 \times 10^{-8}}$	8.17×10^{3}	1.77×10^{6}
$P2$	$[-20, 20]^n$	20	19,000	$\mathbf{3.00 \times 10^{-8}}$	7.78×10^{-6}	1.03
		50	76,250	$\mathbf{1.40 \times 10^{-8}}$	2.09×10^{-2}	1.82
		100	245,000	$\mathbf{1.00 \times 10^{-8}}$	3.70×10^{2}	5.77
	$[20, 60]^n$	20	22,000	$\mathbf{2.12 \times 10^{-6}}$	2.84×10^{-5}	1.59×10^{5}
		50	102,500	$\mathbf{1.20 \times 10^{-8}}$	4.20×10^{-3}	3.30×10^{4}
		100	327,500	$\mathbf{1.00 \times 10^{-8}}$	5.73×10^{2}	6.23×10^{4}
$P3$	$[-20, 20]^n$	20	25,500	$\mathbf{1.03 \times 10^{-6}}$	2.74×10^{-5}	2.18×10^{3}
		50	122,500	$\mathbf{9.53 \times 10^{-7}}$	5.24×10^{-2}	1.01×10^{4}
		100	440,000	$\mathbf{1.25 \times 10^{-6}}$	1.15×10^{5}	3.55×10^{4}
	$[20, 60]^n$	20	28,500	$\mathbf{2.04 \times 10^{-4}}$	6.30×10^{-5}	3.95×10^{7}
		50	137,500	$\mathbf{4.54 \times 10^{-5}}$	5.63×10^{-2}	1.44×10^{8}
		100	485,000	$\mathbf{3.63 \times 10^{-5}}$	3.52×10^{5}	4.08×10^{8}
$P4$	$[-20, 20]^n$	20	38,000	$\mathbf{1.07 \times 10^{-6}}$	5.27×10^{-6}	4.35×10^{1}
		50	185,000	$\mathbf{3.25 \times 10^{-7}}$	9.01×10^{-4}	9.08×10^{1}
		100	657,500	$\mathbf{3.38 \times 10^{-7}}$	6.46×10^{1}	1.76×10^{2}
	$[20, 60]^n$	20	41,500	$\mathbf{5.09 \times 10^{-6}}$	6.56×10^{-6}	7.18×10^{3}
		50	198,750	$\mathbf{1.15 \times 10^{-6}}$	1.10×10^{-3}	1.27×10^{-4}
		100	707,500	$\mathbf{9.60 \times 10^{-7}}$	1.06×10^{2}	2.09×10^{4}
$P5$	$[-20, 20]^n$	20	17,000	$\mathbf{4.00 \times 10^{-8}}$	8.42×10^{-7}	2.29
		50	65,000	$\mathbf{1.20 \times 10^{-8}}$	3.30×10^{-3}	3.31×10^{-2}
		100	200,000	$\mathbf{1.00 \times 10^{-8}}$	1.14	5.20×10^{-2}
	$[20, 60]^n$	20	22,500	$\mathbf{4.00 \times 10^{-8}}$	8.42×10^{-7}	2.29
		50	88,750	$\mathbf{1.60 \times 10^{-8}}$	8.76×10^{-4}	4.71
		100	280,000	$\mathbf{1.10 \times 10^{-8}}$	1.21	1.15×10^{1}
$P6$	$[-20, 20]^n$	20	23,000	$\mathbf{1.00 \times 10^{-8}}$	1.34×10^{-6}	3.69×10^{-2}
		50	90,000	$\mathbf{1.20 \times 10^{-8}}$	1.26×10^{-4}	7.59×10^{-2}
		100	270,000	$\mathbf{1.10 \times 10^{-8}}$	2.46×10^{-2}	1.15×10^{-1}
	$[20, 60]^n$	20	27,000	$\mathbf{1.10 \times 10^{-8}}$	5.83×10^{-7}	1.64×10^{1}
		50	107,500	$\mathbf{1.30 \times 10^{-8}}$	6.22×10^{-5}	1.96×10^{1}
		100	330,000	$\mathbf{1.00 \times 10^{-8}}$	2.07×10^{-2}	2.49×10^{1}

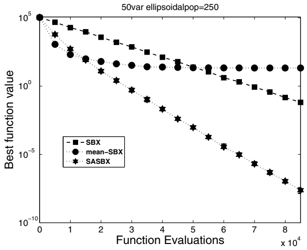


Fig. 7. Population-best objective value for 50-variable problem P1 with symmetric initialization

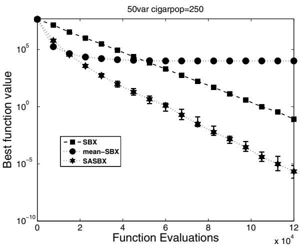


Fig. 8. Population-best objective value for 50-variable problem P3 with symmetric initialization

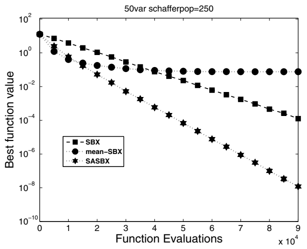


Fig. 9. Population-best objective value for 50-variable problem P6 with symmetric initialization

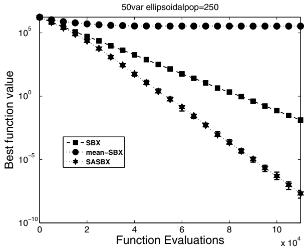


Fig. 10. Population-best objective value for 50-variable problem P1 with one-sided initialization

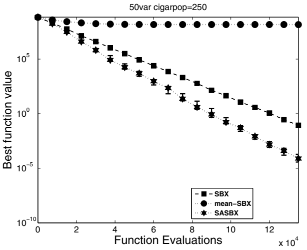


Fig. 11. Population-best objective value for 50-variable problem P3 with one-sided initialization

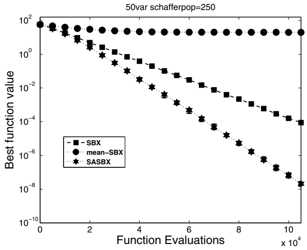


Fig. 12. Population-best objective value for 50-variable problem P6 with one-sided initialization

5 Conclusions

In this paper, we have argued and demonstrated that instead of using a purely parent-centric or a purely mean-centric recombination operator, an adaptive use of them in a real-parameter optimization algorithm is a better overall approach. By modifying the existing simulated binary crossover (SBX) operator with an adaptive parameter that depends on the location of population-best solution compared to the diversity associated with the population, an self-adaptive transition from mean-centric to parent-centric versions of SBX recombination operator has been achieved. On a number of unimodal and multi-modal test problems having 20 to 100 variables, the real-parameter genetic algorithm (RGA) with adaptive SBX has consistently found the global minimum, whereas other algorithms could not solve all problems for the allocated number of function evaluations. This proof-of-principle study shows promise for a more detailed study in which (i) a number of other more complex problems can be taken up, (ii) the principle of self-adaptive transition from parent-centric to mean-centric operations can be extended to multi-objective optimization problems. We are currently pursuing some of these studies, however, this study has shown a clear indication of superiority of an adaptive approach compared to either purely parent-centric or purely mean-centric recombination approaches.

References

1. Deb, K., Agrawal, R.B.: Simulated binary crossover for continuous search space. Complex Systems 9(2), 115–148 (1995)
2. Deb, K., Anand, A., Joshi, D.: A computationally efficient evolutionary algorithm for real-parameter optimization. Evolutionary Computation Journal 10(4), 371–395 (2002)
3. Deb, K., Sindhya, K., Okabe, T.: Self-adaptive simulated binary crossover for real-parameter optimization. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2007), pp. 1187–1194. The Association of Computing Machinery (ACM), New York (2007)
4. Eshelman, L.J., Schaffer, J.D.: Real-coded genetic algorithms and interval-schemata. In: Foundations of Genetic Algorithms 2 (FOGA-2), pp. 187–202 (1993)
5. Hansen, N., Ostermeier, A.: Completely derandomized self-adaptation in evolution strategies. Evolutionary Computation Journal 9(2), 159–195 (2000)
6. Higuchi, T., Tsutsui, S., Yamamura, M.: Theoretical Analysis of Simplex Crossover for Real-Coded Genetic Algorithms. In: Deb, K., Rudolph, G., Lutton, E., Merelo, J.J., Schoenauer, M., Schwefel, H.-P., Yao, X. (eds.) PPSN 2000. LNCS, vol. 1917, pp. 365–374. Springer, Heidelberg (2000)
7. Ono, I., Kobayashi, S.: A real-coded genetic algorithm for function optimization using unimodal normal distribution crossover. In: Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA-7), pp. 246–253 (1997)
8. Voigt, H.-M., Mühlenbein, H., Cvetković, D.: Fuzzy recombination for the Breeder Genetic Algorithm. In: Proceedings of the Sixth International Conference on Genetic Algorithms, pp. 104–111 (1995)

Attribute Reduction in Decision-Theoretic Rough Set Models Using Genetic Algorithm

Srilatha Chebrolu and Sriram G. Sanjeevi

Department of Computer Science and Engineering,
NIT Warangal, AP, India
{srilatha.chebrolu,sgs}@nitw.ac.in

Abstract. Real world data may contain inconsistencies, uncertainty and noise. Rough set model is a mathematical methodology in data analysis to deal with inconsistent and imperfect knowledge. Various probabilistic approaches to rough set model are proposed. Decision-theoretic rough set model (DTRSM) is one of the probabilistic approaches to rough set model. This paper proposes an attribute reduction algorithm in DTRSM, through region preservation. Attribute reduction is the process of identifying and removing redundant and irrelevant attributes from huge data sets, reducing its volume. The reduced data set can be much more effectively analyzed. Attribute reduction in DTRSM through region preservation is an optimization problem, thus Genetic Algorithm (GA) is used to achieve this optimization. Experiment results on discrete data sets are compared with local optimization approach based on discernibility matrix method and has been shown that GA can be effectively and efficiently used to achieve global minimal reduct.

Keywords: Attribute Reduction, Decision-Theoretic Rough Set Model, Genetic Algorithm.

1 Introduction

RST was introduced by Pawlak in 1982. RST is a mathematical methodology in data analysis. RST handles uncertain information in data sets. RST carry through challenging jobs like attribute reduction, identifying patterns in data, computation of attribute relevance and data set characterization [5]. Some of the applications of RST include machine learning, data mining, decision analysis, pattern recognition and knowledge discovery [5]. RST requires rigid participation specifications of including entities into the approximation regions. RST is extended to Probabilistic rough set model (PRSM) by relaxing these strict participation rules, through introducing a pair of threshold parameters α and β, $0 \leq \beta < \alpha \leq 1$ to redefine RST lower and upper approximations [12], [2]. Variable precision rough set model [17], Bayesian rough set model [8] and DTRSM [9], [10], [11] are few variations of PRSM.

DTRSM was proposed by Yao.Y.Y. DTRSM is based on Bayesian decision procedure to compute threshold parameters by minimizing loss, decision costs are taken into account while classifying an equivalence class into a specific region [11].

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 307–314, 2011.
© Springer-Verlag Berlin Heidelberg 2011

DTRSM deals semantic interpretations of threshold parameters and decision rules in PRSM [14]. Various PRSM's can be derived from DTRSM [10], [11]. Three-way interpretation of rules in RST was proposed in [13]. Classification properties like decision-monotonicity, confidence, coverage, generality and costs in DTRSM attribute reduction are discussed in [15].

Attribute reduction in DTRSM can be interpreted as either decision preservation or region preservation [16]. Attribute reduction in DTRSM through decision preservation is the process of finding a minimal subset of conditional attributes that has the same decisions as the entire set of conditional attributes. Discernibility matrix approach is used in [16] to solve this decision problem. Attribute reduction in DTRSM through region preservation is the process of finding a minimal reduct. A minimal reduct is the minimum subset of conditional attributes for which the probabilistic region is the superset of the probabilistic region of the entire set of conditional attributes and for any subset of the reduct, its probabilistic region should be the subset of probabilistic region of the minimal reduct. Finding minimal reduct is an optimization problem. Discernibility matrix approach used in [16], achieves local optimization. We propose a Genetic algorithm (GA) to achieve global optimization as it can efficiently solve optimization problems.

2 Rough Sets and Decision Theoretic Rough Sets

Any real world data can be represented and stored in the form of an information table. Information table is also known as decision table. All rows of the information table called objects or examples makes up knowledge. Entire data is described by a set of properties called conditional attributes. Decision attribute of each object determines the class of the object.

Definition 1. *[5] Information table is defined as a 5 tuple* $S = (U, C, d, V, f)$*, where* U *is the universe of all objects* $\{o_1, o_2, o_3, ...\}$*,* C *is the set of all conditional attributes* $\{c_1, c_2, c_3, ...\}$*,* d *is the decision attribute, satisfying* $d \notin C$*,* $V = \bigcup_{c_i \in C \cup \{d\}} V_{c_i}$*, where* V_{c_i} *is the domain of the attribute* c_i*,* $f : U \times C \rightarrow V$ *is a mapping function, where* $f(o_i, c_j)$ *represents a value for object* o_i *under attribute* c_j *in the domain* V_{c_j}*.*

Definition 2. *[5] The principal idea of RST is indiscernibility relation over any subset of conditional attributes* A*,* $A \subseteq C$*. Indiscernibility relation is defined as, the set of objects characterized by the same values for all the attributes in* A*. Indiscernibility relation is also known as equivalence relation as it satisfies reflexive, symmetric and transitive relations. Indiscernibility relation is denoted by* I_A*.* $I_A = \{(o_i, o_j) \in U \times U \mid f(o_i, a) = f(o_j, a), \forall a \in A\}$*. Incernibility relation over* U *is denoted by* $U|I_A$*,* $U|I_A = \{[o_i]_A \mid o_i \in U\}$ *where* $[o_i]_A$ *is the equivalence class of* o_i *over* A*.*

Definition 3. *[5] Decision class of a decision* $x_i \in V_d$ *is defined as, the set of all objects with* x_i *as the decision. Decision class of* x_i *is denoted by* X_i*,* $X_i = \{o_j | f(o_j, d) = x_i, \forall o_j \in U\}$*. The set of all decision classes, denoted by* D*, where* $D = \{X_1, X_2, X_3, ...\}$ *partitions* U*.*

Definition 4. [5] *Rough set of a decision class X_i over any subset of conditional attributes $A \subseteq C$, is defined by lower approximation and upper approximation. Lower approximation of X_i over A is defined as, the set of objects that are certainly of decision class X_i. Upper approximation is defined as, the set of objects that may be of decision class X_i.*

$$\underline{apr}^A(X_i) = \bigcup_{[o_j]_A \in U|I_A} \{[o_j]_A \mid [o_j]_A \subseteq X_i\}$$
$$\overline{apr}^A(X_i) = \bigcup_{[o_j]_A \in U|I_A} \{[o_j]_A \mid [o_j]_A \cap X_i \neq \phi\}$$

Definition 5. [16] *The probability of $o_j \in U$ over any subset of conditional attributes $A \subseteq C$, is of decision class X_i is defined as, the ratio of the number of elements that are correctly classified as of X_i to the number of total elements in $[o_j]_A$. The classification probability is denoted by $P(X_i \mid [o_j]_A)$.*

$$P(X_i|[o_j]_A) = \frac{\mid X_i \cap [o_j]_A \mid}{\mid [o_j]_A \mid}$$

Definition 6. [16] *Probabilistic lower approximation of decision class X_i over a subset of conditional attributes $A \subseteq C$, in the interval (α, β), is defined as, the set of objects that are of decision class X_i with probability $\geq \alpha$*

$$\underline{apr}^A_{(\alpha,\beta)}(X_i) = \bigcup_{[o_j]_A \in U|I_A} \{[o_j]_A \mid P(X_i \mid [o_j]_A) \geq \alpha\}$$

Probabilistic upper approximation of decision class X_i over any subset of conditional attributes $A \subseteq C$, in the interval (α, β), is defined as, the set of objects that are of decision class X_i with probability $> \beta$

$$\overline{apr}^A_{(\alpha,\beta)}(X_i) = \bigcup_{[o_j]_A \in U|I_A} \{[o_j]_A \mid P(X_i \mid [o_j]_A) > \beta\}$$

Definition 7. [16] *Universe U is partitioned into 3 disjoint probabilistic regions. The (α, β) - positive region, (α, β) - boundary region and (α, β) - negative region. Probabilistic positive region of decision class X_i over any subset of conditional attributes $A \subseteq C$, in the interval (α, β), is defined as, the set of objects of decision class X_i with probability $\geq \alpha$*

$$POS^A_{(\alpha,\beta)}(X_i) = \bigcup_{[o_j]_A \in U|I_A} \{[o_j]_A \mid P(X_i \mid [o_j]_A) \geq \alpha\}$$

Probabilistic boundary region of decision class X_i over any subset of conditional attributes $A \subseteq C$, in the interval (α, β), is defined as, the set of objects of decision class X_i with probability in between α and β

$$BND^A_{(\alpha,\beta)}(X_i) = \bigcup_{[o_j]_A \in U|I_A} \{[o_j]_A \mid \beta < P(X_i \mid [o_j]_A) < \alpha\}$$

Probabilistic negative region of decision class X_i over any subset of conditional attributes $A \subseteq C$, in the interval (α, β), is defined as, the set of objects of decision class X_i with probability $\leq \beta$

$$NEG^A_{(\alpha,\beta)}(X_i) = \bigcup_{[o_j]_A \in U|I_A} \{[o_j]_A \mid P(X_i \mid [o_j]_A) \leq \beta\}$$

Probabilistic non-negative region of decision class X_i over any subset of conditional attributes $A \subseteq C$, in the interval (α, β), is defined as, the union of probabilistic positive region and probabilistic boundary region.

$$\overline{NEG}^A_{(\alpha,\beta)}(X_i) = POS^A_{(\alpha,\beta)}(X_i) \cup BND^A_{(\alpha,\beta)}(X_i)$$

Probabilistic positive region over all decision classes, D is defined as

$$POS^A_{(\alpha,\beta)}(D) = \bigcup_{X_i \in D} POS^A_{(\alpha,\beta)}(X_i)$$

Definition 8. *[16] As probabilistic regions are non-monotonic, a minimal reduct in DTRSM through positive region preservation, denoted by A, is defined as the subset of conditional attributes, $A \subseteq C$ with maximum probabilistic positive region among all the subsets of C.*

$$\arg\max_{A \subseteq C}\{POS^A_{(\alpha,\beta)}(D)\}$$

A minimal reduct in DTRSM through any region preservation can be defined in a similar way. The next section describes a local optimization technique to attain local minimal reduct.

3 Local Optimization Approach Using Discernibility Matrix Method

Discernibility matrix is introduced by Skowron and Rauszer [7]. Local minimal reduct is found using this approach [16]. Objects in U form rows and columns of the matrix. Element of the matrix is defined as the set of all attributes that discerns row object from column object, with different probabilistic regions. Matrix element for positive region preservation is given as,

$$c_{ij} = \begin{cases} \{a \in C | f(o_i, a) \neq f(o_j, a)\}, POS_{(\alpha,\beta)}([o_i]_C) \neq POS_{(\alpha,\beta)}([o_j]_C) \\ \phi, POS_{(\alpha,\beta)}([o_i]_C) = POS_{(\alpha,\beta)}([o_j]_C) \end{cases}$$

Attribute reduct sets with the same probabilistic region as C, are the prime implicants of the discernibility function. Discernibility function for positive region preservation is defined as

$$f(M_{POS_{(\alpha,\beta)}}) = \bigwedge\{\bigvee M_{POS_{(\alpha,\beta)}}\}, \text{ where } M_{POS_{(\alpha,\beta)}} = \{c_{ij} | 1 \leq i < j \leq n\}$$

Local minimal reduct is the subset of the attribute reduct set with probabilistic region as superset of the probabilistic region of C. If there is no such subset, then attribute reduct itself is the local minimal reduct. The next section describes a global optimization technique to achieve global minimal reduct.

4 Global Optimization Approach Using Genetic Algorithm

GA provides a methodology to solve optimization problems. GA is motivated by biological evolution [3] and is stochastic in directing a huge search space. The algorithm to find minimal reduct in DTRSM, through region preservation using GA, is as shown in Algorithm 1. The total number of candidate reducts is $2^n - 1$, where n is $|C|$.

Algorithm 1. Finding minimal reduct in DTRSM, through region preservation, using GA

```
Input: Population type, population size, creation function, scaling function, selection function, elite count, crossover rate, fitness function, crossover function, mutation function, iteration count
Output: Minimal reduct - a candidate reduct with maximum fitness value.
begin
  1: Generate initial population randomly, of given population size from a collection of $2^n - 1$ possible candidate reducts, where n is the total number of conditional attributes in the given information table.
  2: Evaluate fitness value, for each candidate reduct by using the fitness function given in equation (2), if attribute reduction is through positive region preservation. Use a corresponding fitness function, if attribute reduction is through any other region preservation.
  3: Until iteration count is elapsed, repeat steps 4-6.
  4: Generate next population by applying crossover and mutation operations.
  5: Evaluate fitness value for each candidate reduct of the population.
  6: Decrement iteration count by 1
  7: Output the candidate reduct, that has the maximum fitness value from the current population.
end
```

4.1 Representation

Each candidate reduct is represented as a chromosome by a bit string of length $\mid C \mid$. Let C = $\{c_1, c_2, ..., c_n\}$, be the set of conditional attributes, then a 1 in the bit string at position i represents c_i is present in the candidate reduct and a 0 in the bit string at position i represents c_i is not present in the candidate reduct.

4.2 Fitness Function

Fitness function of a candidate reduct A is denoted by $f(ch_A)$. Candidate reduct with a minimum number of attributes and with maximum number of objects in its probabilistic region will have maximum fitness value, and such candidate reducts will survive through GA iterations. The candidate reduct with maximum fitness value will be selected as the minimal reduct. For attribute reduction through positive region preservation, fitness value is calculated only for those candidate reducts for which,

$$POS^{A}_{(\alpha,\beta)}(D) \supseteq POS^{C}_{(\alpha,\beta)}(D) \tag{1}$$

Fitness function for attribute reduction through positive region preservation, is defined as

$$f(ch_A) = \begin{cases} \dfrac{\mid C \mid - \mid A \mid}{|C|} + |POS^{A}_{(\alpha,\beta)}(D)| & \text{if (1) holds} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where ch_A represents a chromosome. Fitness function for attribute reduction through any region preservation, can be defined in a similar way.

4.3 Proof of Fitness Function

The correctness of fitness function for attribute reduction through positive region preservation, defined in equation (2) is proved in Theorem 1.

Theorem 1. *Candidate reduct with maximum fitness value is the minimal reduct.*

Proof. Consider two candidate reducts, A and B satisfying constraint (1) and let $f(ch_A) > f(ch_B)$. *We have to prove that A is a minimal reduct compared to B. So it is sufficient to prove that the probabilistic positive region of A has more objects compared to the probabilistic positive region of B, i.e.,* $|POS^{A}_{(\alpha,\beta)}(D)| > |POS^{B}_{(\alpha,\beta)}(D)|$. *This has to be proven for all the 3 possible cases of* $|A| > |B|, |A| = |B|, |A| < |B|$.

Case 1. when $|A| > |B|$
$$\Rightarrow \frac{|C|-|A|}{|C|} < \frac{|C|-|B|}{|C|} \tag{3}$$

$$Given f(ch_A) > f(ch_B)$$
$$\Rightarrow \frac{|C|-|A|}{|C|} + |POS^{A}_{(\alpha,\beta)}(D)| > \frac{|C|-|B|}{|C|} + |POS^{B}_{(\alpha,\beta)}(D)|$$
$$\Rightarrow |POS^{A}_{(\alpha,\beta)}(D)| > |POS^{B}_{(\alpha,\beta)}(D)| \textit{ from in-equation (3)}$$

Case 2. when $|A| = |B|$
$$\Rightarrow \frac{|C|-|A|}{|C|} = \frac{|C|-|B|}{|C|} \tag{4}$$

$$Given\ f(ch_A) > f(ch_B),$$
$$\Rightarrow \frac{|C|-|A|}{|C|} + |POS^{A}_{(\alpha,\beta)}(D)| > \frac{|C|-|B|}{|C|} + |POS^{B}_{(\alpha,\beta)}(D)|$$
$$\Rightarrow |POS^{A}_{(\alpha,\beta)}(D)| > |POS^{B}_{(\alpha,\beta)}(D)|\ from\ equation\ (4)$$

Case 3. when $|A| < |B|$
For all the possible candidate reducts,

$$0 < \frac{|C|-|A|}{|C|} < 1\ and\ 0 < \frac{|C|-|B|}{|C|} < 1 \tag{5}$$

$$|POS^{A}_{(\alpha,\beta)}(D)| \in \mathbb{N}\ and\ |POS^{B}_{(\alpha,\beta)}(D)| \in \mathbb{N} \tag{6}$$

where $\mathbb{N}$ *is the set of Natural numbers.*

$$Given\ f(ch_A) > f(ch_B)$$

$$\Rightarrow \frac{|C|-|A|}{|C|} + |POS^{A}_{(\alpha,\beta)}(D)| > \frac{|C|-|B|}{|C|} + |POS^{B}_{(\alpha,\beta)}(D)|$$

$\Rightarrow |POS^{A}_{(\alpha,\beta)}(D)| > |POS^{B}_{(\alpha,\beta)}(D)|$ *from constraints (5) and (6).*

The theorem is proved for all the 3 possible cases.

Correctness of the fitness function for attribute reduction through any region preservation, can be proved in a similar way.

5 Experiments

The proposed algorithm is evaluated on 3 discrete data sets, mainly Wine [1], Turbine [4] and an example data set described in [6]. All these 3 data sets contain inconsistent data. Table 1 shows the experimental results of local optimization approach based on discernibility matrix method. In this approach attribute sets with the same probabilistic region as C are identified and a local optimal reduct with respect to an identified attribute set is found by checking all its subsets. Table 2 shows GA parameter settings to perform experiments on all the 3 data sets. The algorithm is implemented using MATLAB, minimization of $-f(ch)$ is used to achieve maximization of $f(ch)$. Table 3 summarizes experimental results of global optimization approach using GA. GA is found to achieve global minimal reduct in all the 3 data sets.

Table 1. Experiment results of Local optimization approach based on Discernibility matrix method

Data set	α	β	Attribute sets with the same probabilistic region as C does	Local minimal reduct
Wine [1]	0.75	0.64	$\{c_1, c_3, c_5, c_8\}$	$\{c_1, c_3, c_5, c_8\}$
Turbine [4]	0.6	0.4	$\{c_4, c_5, c_7\}$, $\{c_3, c_4, c_5\}$, $\{c_1, c_4, c_5\}$	$\{c_3, c_4, c_5\}$
Example [6]	0.5	0.3	$\{c_1, c_3, c_4, c_5\}$, $\{c_1, c_2, c_3\}$	$\{c_1, c_3, c_4, c_5\}$

Table 2. GA parameter settings

Population size	Creation function	Scaling function	Selection function	Crossover fraction	Mutation function	Crossover function	Stopping criteria
20	Uniform	Rank	Stochastic uniform	0.8	Gaussian	Scattered	Iteration count:100

Table 3. Experiment results of GA with the proposed fitness function

Data set	α	β	No. of objects in positive region of C	No. of objects in positive region of minimal reduct	Fitness Value of minimal reduct	Reduced attributes	Global minimal reduct	Reduct-ion (%)
Wine [1]	0.75	0.64	46	52	52.625	$\{c_1, c_2, c_3, c_6, c_7\}$	$\{c_4, c_5, c_8\}$	62.5
Turbine [4]	0.6	0.4	38	40	40.625	$\{c_1, c_3, c_6, c_7, c_8\}$	$\{c_2, c_4, c_5\}$	62.5
Example [6]	0.5	0.3	25	28	28.4	$\{c_1, c_3\}$	$\{c_2, c_4, c_5\}$	40

6 Conclusion

Attribute reduction is a pre-process step in data analysis, removing superfluous data. In Decision-theoretic rough set model (DTRSM), attribute reduction can be achieved either by region preservation or by decision preservation. Discernibility matrix approach, achieves global minimal reducts through decision preservation, but it achieves only local minimal reducts through region preservation because of non-monotonic property of probabilistic regions in DTRSM. Thus we have proposed a GA solution to achieve global minimal reduct in DTRSM through region preservation. Both the approaches have been tested across 3 different data sets Wine [1], Turbine [4] and an example data set described in [6] and the results shows that discernibility matrix approach achieves local optimization, where as the proposed GA achieves global optimization and is found to be efficiently working.

References

1. Beynon, M.: Reducts within the variable precision rough sets model: A further investigation. European Journal of Operational Research, 592–605 (2001)
2. Dun, L., Huaxiong, L., Xianzhong, Z.: Two decade's research on decision-theoretic rough sets. In: ICCI 2010, pp. 968–973 (2010)
3. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Pearson Education (2009)
4. Lin, L., Guang-hua, X.: Reduction of rough set attribute based on immune clone selection. Front. Mech. Eng. 4, 413–417 (2006)
5. Pawlak, Z., Grzymala-Busse, J., Slowinski, R., Ziarko, W.: Rough sets. Communications of the ACM 8, 85–99 (1995)
6. Skowron, A., Grzymalla-Busse, J.: From rough set theory to evidence theory. In: Advances in the Dempster-Shafer Theory of Evidence. Wiley, New York (1994)
7. Skowron, A., Rauszer, C.: The discernibility matrices and functions in information systems. In: Intelligent Decision Support Handbook of Applications and Advance of the Rough Sets Theory (1992)
8. Slezak, D.: Rough sets and bayes factor. T. Rough Sets, 202–229 (2005)
9. Yao, Y.: Probabilistic approaches to rough sets. Expert Systems 20, 287–297 (2003)
10. Yao, Y.: Decision-Theoretic Rough Set Models. In: Yao, J., Lingras, P., Wu, W.-Z., Szczuka, M.S., Cercone, N.J., Ślęzak, D. (eds.) RSKT 2007. LNCS (LNAI), vol. 4481, pp. 1–12. Springer, Heidelberg (2007)
11. Yao, Y.: Decision-theoretic rough set models (dtrsm). Computer Science 35(8A), 7–8 (2008)
12. Yao, Y.: Probabilistic rough set approximations. Int. J. Approx. Reasoning, 255–271 (2008)
13. Yao, Y.: Three-Way Decision: An Interpretation of Rules in Rough Set Theory. In: Wen, P., Li, Y., Polkowski, L., Yao, Y., Tsumoto, S., Wang, G. (eds.) RSKT 2009. LNCS, vol. 5589, pp. 642–649. Springer, Heidelberg (2009)
14. Yao, Y.: Two semantic issues in a probabilistic rough set model. Fundamenta Informaticae, 1–17 (2010)
15. Yao, Y., Zhao, Y.: Attribute reduction in decision-theoretic rough set models. Inf. Sci., 3356–3373 (2008)
16. Zhao, Y., Wong, S.K.M., Yao, Y.: A Note on Attribute Reduction in the Decision-Theoretic Rough Set Model. In: Peters, J.F., Skowron, A., Chan, C.-C., Grzymala-Busse, J.W., Ziarko, W.P. (eds.) Transactions on Rough Sets XIII. LNCS, vol. 6499, pp. 260–275. Springer, Heidelberg (2011)
17. Ziarko, W.: Variable precision rough set model. J. Comput. Syst. Sci., 39–59 (1993)

A Study of Decision Tree Induction for Data Stream Mining Using Boosting Genetic Programming Classifier

Dirisala J. Nagendra Kumar[1], J.V.R. Murthy[2],
Suresh Chandra Satapathy[3], and S.V.V.S.R. Kumar Pullela[4]

[1] BVRICE, Bhimavaram, India
[2] JNTUCE, Kakinada, India
[3] ANITS, Visakhapatnam, India
[4] V.S. Lakshmi Engineering College, Kakinada, India
{nagendrakumardj,mjonnalagedda,sureshsatapathy,
ravipullela}@gmail.com

Abstract. Genetic Programming is an evolutionary soft computing approach. Data streams are the order of the day input mechanisms. Here is a study of GP Classifier on Data Streams. GP classification performance is compared to that of other state-of-the-art data mining and stream classification approaches. Boosting is a machine learning meta-algorithm for performing supervised learning. A weak learner is defined to be a classifier which is only slightly correlated with the true classification (it can label examples better than random guessing). In contrast, a strong learner is a classifier that is arbitrarily well-correlated with the true classification. Boosting combines a set of weak learners to create a strong learner. It is observed that the Boosting GP approach is beating Boosting Naïve Bayes classification. Hence it is found that GP is a competent algorithm for Data Stream classification.

1 Introduction

Many organizations are being dumped with tremendous amount of continuous flow of data, due to a sequence of events from different locations of the organization. Telephone records, credit/debit card transactions, sensor networks, network event logs, web log data, online sales transactions are some examples of data streams. Traditional approach mining data is known batch processing, as it assumes data as a static entity. Now data streams stresses the need of online and incremental data mining techniques, of course should be able to deal with concepts drifts in some cases.

Data stream classification is studied in [15-20]. Mohammad M. Masud et. al. studied Mine Class Algorithm for automatic detection of a novel class in presence of concept-drift [15]. Gianluigi Folino et. al. has studied a StreamGP approach with adaptive boosting ensemble algorithm for classifying homogeneous distributed data streams [16]. Hussein A. Abbass et. al. made a detailed study of online adaption in learning classifier systems for stream data mining based on Genetic Algorithms [17]. Yi Zhnag and Xiaoming Jin built an ensemble classification technique on data streams [18]. New ensemble methods for evolving data streams are studied by Albert

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 315–322, 2011.
© Springer-Verlag Berlin Heidelberg 2011

Bifet et. al. [20]. Things are moving faster. Wenyan Wu and Le Gruenwald studied various issues involved in simultaneous mining of multiple data streams [19].

Most of the work on classification concentrates on binary classification problems. Traditionally Maximum Likelihood Classifier (MLC) [10], Bayesian networks [10], and Neural networks (NN) [11] are the most successful approaches for multi-class classification.

Genetic Programming (GP) is a stochastic approach, derived from Genetic Algorithms (GA), to solve various computer related problems by automatically constructing programs simulating the biological evolution [8]. GP is a nice approach for solving the binary and multi-class classification problems. It guarantees good classification accuracy if enough training time is given to evolve a higher accuracy GP classifier [2]. An attempt is made to reduce this training time to a reasonable degree. The goals that are tried to meet are Simplicity, Scalability, and high accuracy.

The GP classifier has to find fitness for all fitness cases, which may not be stored in main memory for larger datasets. In order to achieve scalability, the size of training data set sampled at a time is restricted to a portion of main memory available. Topon Kumar Paul, and Hitoshi Iba [5] implemented the ensemble approach of Boosting based on GP and called it “a majority voting genetic programming classifier”.

T. Loveard and V. Ciesielski [1] proposed five alternative methods to perform GP-based multi-class classification, viz., Binary decomposition, Static range selection, dynamic range selection, class enumeration and evidence accumulation.

J. K. Kishore et al.[2] modeled the n-class pattern classification problem as an n two-class problems. A Genetic programming classifier expression (GPCE) is evolved as a discriminant function for each class. Each GPCE recognizes data samples belonging to its own class and rejects samples belonging to other classes. In [2]-[4], [14] the authors designed a classifier with n binary-trees for the n-class classification problem.

D.P. Muni et al. [3] improved the approach of J.K. Kishore et al.[2] by generating the classifier in one pass. D.P. Muni et al. extended their earlier work to suit for feature selection (FS) in [4], proposing a wrapper approach for FS.

Topan Kumar Paul, and Hitoshi Iba [5] proposed a majority voting technique, which evolves multiple GP rules and apply those rules to test samples to determine their labels and count their votes in favor of a particular class. Then the sample is assigned to the class that gets the highest number of votes in favor of it.

T. Loveard and V. Ciesielski[1] used the total training set as exemplar set. In [3], [4], D.P. Muni et al. used step-wise learning, which takes a smaller exemplar set initially, and gradually increases the exemplar set to the whole training set.

2 Data Streams

The recent advances in hardware and software have enabled the capture of different measurements of data in a wide range of fields. These measurements are generated continuously and in a very high fluctuating data rates. Examples include sensor networks, web logs, and computer network traffic. The storage, querying and mining of such data sets are highly computationally challenging tasks. Mining data streams is concerned with extracting knowledge structures represented in models and patterns in non stopping streams of information. The research in data stream mining has gained a high attraction due to the importance of its applications and the increasing generation of streaming information. Applications of data stream analysis can vary from critical scientific and astronomical applications to important business and financial ones. Algorithms, systems and frameworks that address streaming challenges have been developed over the past decade. There is a real need inspired by the potential applications in astronomy and scientific laboratories as well as business applications.

3 Our Boosting Approach for Data Stream Classification

Once a program prediction was obtained with a search for GP, it can be used directly for classification, or continue to create programs to combine them into a more efficient solution later . There are several ways to combine systems of statistical predictions. For example, one can use a voting system for the results of several classifiers.

In the case of a classification problem with n classes, there are several approaches to the resolution. The three most common approaches are:

1. Develop a single classification that gives, as output, the class of the new sample as input.
2. Develop n classifiers. Each classifier is responsible for recognizing a particular class.
3. Develop a classifier for each pair of classes (the number of classifiers is given by the sum iterative i, i = 1 to i = n - 1). Each classifier is responsible to decide between two classes in particular.

The method 2, a classifier that uses samples per class, is the best approach [23]. So this variant is to be used by our algorithm. With this method, the classifiers obtained by the GP must have some type of output value. Two approaches were again proposed:

1. Binary classifier: the result is 0, it predicts that the sample is not part of the class, or if one predicts that the sample belongs to the class.
2. Classifier output continuous, the result is a decimal value (eg between 0.0 and 1.0) that represents the confidence with which the classifier combines the sample with the designated class.

When a new sample is introduced, each classifier must predict whether the sample belongs to the class for which it was trained. The combined classifier has the output value that determines the largest class of new sample. In the event of a tie, the classifier that has the highest probability will be identified as the class.

We will integrate the meta algorithm for boosting the evolutionary process of the GP. We developed boosting algorithm with n-classifiers (one for each class) and each classifier predicting the confidence with which the classifier is assigned the class. Several studies on the implementation of a method for boosting the GP have reported significant gains in terms of recognition performance and computation time of the algorithm [21-22]. The integration of the principles of boosting even within the GP process allows greater economy of resources. Here is our Boosting GP approach in the form of pseudo code:

```
C = number of classes of the problem
P = number of necessary programs for boosting
Training set T = all training data available
N = total number of samples in T
For all Cj (j = 1 to C)
        Empty the population program POP
        Initialize the "weight" of each sample W with Wi = 1 /N
        For all Pk (k = 1 to P)
                If POP is empty, fill POP with a new set of programs .
                Changing a program that recognizes the class C (the calculation of fitness uses the weight Wi of each sample to classify), using T and POP.
                Calculate the error of the best program, $E_{jk}$ on the training set, a factor $\alpha_{jk}$ and then the weight of each sample $W_i$ learning AdaBoost method.
        End for P
End for C
```

A sample can be classified using the strongest response in a weighted sum of the outputs of programs by class (using equation 1). At the end of the routine, we obtain P*C programs (P programs for each class C). The classification score for class j is obtained by the weighted sum by α_{jk} output of each program jk. The class that scores the highest indicates the class of the given sample:

$$\max \left(\sum_{k=1}^{P} (a_{jk} * \alpha_{jk}) \right) \tag{1}$$

A gain of boosting integrated computational resources compared to the standard boosting GP is obtained as the GP does not need to start changing to zero when they are looking for a successful program on the training set new mark weight. It is likely that a successful program for all new weighting already exists in the population developed during the previous search.

4 Fitness Function for Our Boosting GP Approach

The fitness is the measure of GP program performance in the prediction of output values from input sample. It is therefore an indication of relevance of the program for solving the problem represented by the database of learning. Fitness is a numeric value, allowing us to compare the performance of programs. This is the criteria we will use to select programs in the population to transform.

Fitness function is the result of classifications on the training data. This function compares the value of predicted class and actual class provided in the training data. The fitness function depends on the approach used in the combination of classifiers.

1. For a single classifier approach, fitness is simply the number of correct predictions of the program. This value can be normalized (between 0.0 and 1.0) by dividing by the number of matching samples in the data set.
2. In the case of an approach to a sample class classifier, the classifier is the recognition of a particular class. The calculation of the fitness depends on the classifier chosen:

   a) Binary Classifier: As the single classifier approach.
   b) Classifier continuous output: the output of the program P is a value of limited trust between -1.0 and 1.0. Fitness is calculated from the sum of S values of confidence of Pi for each sample, depending on the class C provided by the training data set.

$$S = \sum_{j} P(i) * C(i) \tag{2}$$

   C (i) is 1.0 if the sample i belongs to the class recognized and -1.0 otherwise. Finally, fitness is the sum of S values, normalized between 0.0 and 1.0.
   c) Classifier output continues to boosting algorithm built: the technique is essentially the same as b, but the weight W of training samples is taken into account:

$$S = \sum_{j} P(i) * W(i) * C(i) \tag{3}$$

As the weight of the samples is also normalized (total weight is 1.0), the sum S can be normalized in the same way as in b. So for a classification problem, the more fit, the more the program is effective. A perfect prediction rate is obtained when the fit is 1.0. Every Genetic Programming approach needs some parameters to be specified. In our approach, the GP parameters we used are given in Table 1.

Table 1. The default GP parameters used for GP Classifier Construction

Parameter	Values
Population size	100
Maximum depth	5
Stopping Criteria	Fitness=99%, Max.Generations=100, Max. Time=5 min.
Population Initialization	Ramped-half-and-half
Selection	Roulette wheel
GP operator proportions	Crossover=90%, Mutation=7%, New Program=3%

5 Results

The data on which the classifiers are executed are 2-class Random Tree and 5-class Random tree each with 10Millions rows and the evaluation is through interleaved test then train evaluation. The result of GP classification on the above datasets is as follows:

Table 2. The time taken in secs and classifier accuracy in % of various classifiers on 2-class Random tree

Classifier	Functions	Time in sec.	Accuracy %
AdaBoost M1+ NB	--	1192.05	72.397
GP	+, -, /, *, If, >, <, Pow, &, |, Max, Min, Exp, Log	145.33	69.201
GP	+, -, *, /	119.278	68.877
GP	+, -, *, /, Sq, Sqrt, Pow, Exp, Log	152.304	69.202
GP	+, -, *, /, Sq, Sqrt, Pow, Exp, Log, Min, Max	116.377	68.629
GP	Sq, Sqrt, Pow, Exp, +, -, *, /	173.848	72.533
GP	If, <, >	172.802	71.685
GP	If, If3, <, >	130.214	71.114
GP	If, <, >, !, &, |	118.826	73.27
GP	If, <, >, !, &, |, Xor	86.83	67.331
GP	If, If3, <, >, !, &, |	79.763	57.815
GP	If, If3, <, >, !, &, |, Xor	129.855	68.071

Table 3. The time taken in secs and classifier accuracy in % of various classifiers on 5-class Random tree

Classifier	Functions	Time in sec.	Accuracy %
Adaboost M1+ NB	--	919.532	54.841
GP	+, -, /, *, If, >, <, Pow, &, |, Max, Min, Exp, Log	529.92	50.396
GP	+, -, *, /	749.273	54.585
GP	+, -, *, /, Sq, Sqrt, Pow, Exp, Log	555.551	46.514
GP	+, -, *, /, Sq, Sqrt, Pow, Exp, Log, Min, Max	484.773	50.028
GP	Sq, sqrt, Pow, Exp, +, -, *, /	624.191	52.039
GP	If, <, >	415.321	49.939
GP	If, If3, <, >	693.159	51.771
GP	If, <, >, !, &, |	111.026	33.428
GP	If, <, >, !, &, |, Xor	508.797	50.781
GP	If, If3, <, >, !, &, |	366.789	50.193
GP	If, If3, <, >, !, &, |, Xor	496.551	50.333

In case of the above 2-class Random tree dataset, Boosting GP with If, <, > functions classifying with 72.533% accuracy is better than that of the combination of AdaBoostM1 and Naïve Bayes Classification with 72.397% accuracy. And for 5-class Random tree dataset, Boosting GP with functions +, -, *, / classifying with 54.585% accuracy is almost equal to the combination of Ada boost M1 and Naïve Bayes classifier with 54.841% accuracy. The only disadvantage is that there is no single combination of GP functions and parameters suitable for all datasets. Hence in general, our Boosting GP is a good candidate for Stream classification and may be suitable for further work.

6 Conclusion

It is found that Boosting GP Classifier is a competent approach for classifying data streams. Even then, the issue is changing accuracy with functions and GP parameters. Our next goal is to improve the GP approach in two respects: accuracy and reducing execution time. Improving GP functions (crossover, mutation, selection, etc.) may result in better accuracy. Applying some statistical method like PCA as a preprocessing step and applying some clustering, like Expectation Maximization clustering, may make the approach faster. Our further research work concentrates in the above directions.

References

1. Loveard, T., Ciesielski, V.: Representing classification problems in genetic programming. In: Proc. Congr. Evolutionary Computation, May 27-30, pp. 1070–1077 (2001)
2. Kishore, J.K., Patnaik, L.M., Mani, V., Agrawal, V.K.: Application of genetic programming for multicategory pattern classification. IEEE Transaction on Evolutionary Computation 4, 242–258 (2000)
3. Muni, D.P., Pal, N.R., Das, J.: A novel approach for designing classifiers using genetic programming. IEEE Trans. Evolut. Comput. 8(2), 183–196 (2004)
4. Muni, D.P., Pal, N.R., Das, J.: Genetic programming for simultaneous feature selection and classifier design. IEEE Transactions on Systems, Man, and Cybernetics, Part B 36(1), 106–117 (2006)
5. Paul, T.K., Iba, H.: Prediction of Cancer class with Majority Voting Genetic Programming Classifier Using Gene Expression Data. IEEE/ACM Trans. on Computational Biology and Bioinformatics 6(2), 363–367 (2009)
6. Han, J., Kamber, M.: Data Mining Concepts and Techniques, 2nd edn. Elsevier (2006)
7. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading (1989)
8. Koza, J.R.: Genetic Programming: On the programming of Computers by Means of Natural Selection. MIT Press, Cambridge (1992)
9. Poli, R., Langdon, W.B., McPhee, N.F.: A field guide to Genetic Programming (March 2008), `http://www.gp-field-guide.org.uk`
10. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. John Wiley and Sons (2001)

11. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representation by error propagation. In: Rumelhart, D.E., McClelland, J.L. (eds.) Parallel Distributed Processing. MIT Press (1986)
12. Breiman, L.: Bagging predictors. Machine Learning 24, 123–140 (1996)
13. Tan, P.-N., Steinbach, M., Kumar, V.: Introduction to Data Mining. Person Education (2006)
14. Nagendra Kumar, D.J., Satapathy, S.C., Murthy, J.V.R.: A scalable genetic programming multi-class ensemble classifier. In: World Congress on Nature & Biologically Inspired Computing, NaBIC 2009, pp. 1201–1206 (2009), doi:10.1109/NABIC.2009.5393788
15. Masud, M.M., Gao, J., Khan, L., Han, J., Thuraisingham, B.: Integrating Novel Class Detection with Classification for Concept-Drifting Data Streams. In: Buntine, W., Grobelnik, M., Mladenić, D., Shawe-Taylor, J. (eds.) ECML PKDD 2009. LNCS, vol. 5782, pp. 79–94. Springer, Heidelberg (2009)
16. Folino, G., Pizzuti, C., Spezzano, G.: An Adaptive Distributed Ensemble Approach to Mine Concept-Drifting Data Streams. In: ICTAI 2007, Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence, vol. 02 (2007)
17. Abbass, H.A., Bacardit, J., Butz, M.V., Llorà, X.: Online Adaptation in Learning Classifier Systems: Stream Data Mining (2004)
18. Zhang, Y., Jin, X.: An automatic construction and organization strategy for ensemble learning on data streams. ACM SIGMOD Record Homepage Archive 35(3) (September 2006)
19. Wu, W., Gruenwald, L.: Research issues in mining multiple data streams. In: StreamKDD 2010, Proceedings of the First International Workshop on Novel Data Stream Pattern Mining Techniques (2010)
20. Bifet, A., Holmes, G., Pfahringer, B., Kirkby, R., Gavaldà, R.: New ensemble methods for evolving data streams. In: 15th ACM SIGKDD Intl. Conference on Knowledge Discovery and Data Mining (KDD 2009), Paris, France (June 2009)
21. Folino, G., Pizzuti, C., Spezzano, G.: Boosting Technique for Combining Cellular GP Classifiers. In: Keijzer, M., O'Reilly, U.-M., Lucas, S., Costa, E., Soule, T. (eds.) EuroGP 2004. LNCS, vol. 3003, pp. 47–56. Springer, Heidelberg (2004)
22. Paris, G., Robilliard, D., Fonlupt, C.: Genetic Programming with Boosting for Ambiguities in Regression Problems. In: Ryan, C., Soule, T., Keijzer, M., Tsang, E.P.K., Poli, R., Costa, E. (eds.) EuroGP 2003. LNCS, vol. 2610, pp. 183–193. Springer, Heidelberg (2003)
23. Teredesai, A., Govindaraju, V.: Issues in Evolving GP based Classifiers for a Pattern Recognition Task. In: Proceedings of the 2004 IEEE Congress on Evolutionary Computation, pp. 509–515. IEEE Press (2004)

Bi-criteria Optimization in Integrated Layout Design of Cellular Manufacturing Systems Using a Genetic Algorithm

I. Jerin Leno[1], S. Saravana Sankar[2], M. Victor Raj[3], and S.G. Ponnambalam[4]

[1] Sardar Raja College of Engineering, Alengulam, Tamil Nadu, India
[2] Kalasalingam University, Krishanankoil, Tamil Nadu, India
[3] Dr.Sivanthi Aditanar College of Engineering, Tiruchendur, India
[4] Department of Mechatronics, Monash University, Petaling Jaya, Malaysia
lenojerin@yahoo.co.in

Abstract. Traditionally the design of the physical layout of the manufacturing system and that of the material flow path and material handling system are carried out in isolation. In this work, an attempt was made on the integrated layout design, that is, to concurrently design the physical layout and the material handling system using a Genetic Algorithm-based methodology. The proposed algorithm was employed to simultaneously optimize two contradicting objectives viz., 1. Total material handling cost 2. Distance-weighted cost of closeness rating score. The algorithm was tested on four different benchmark layouts and with different initial problem data sets. It was found that the proposed algorithm is able to produce satisfactory solutions consistently within a reasonable computational limit.

Keywords: Integrated Layout Design, Genetic Algorithm, Multi-objective optimization.

1 Introduction

Facility layout design is an important issue for any industry, as a poor layout affects overall efficiency of the total production system. Traditionally, the layout design is carried out in a sequential manner involving two major steps [1, 2], viz., (1) Cell System Layout (CSL) design wherein the exact location, orientation, position of input/output stations of each cell are determined, (2) Material Handling System (MHS) design wherein the flow path of the materials between input and output stations of different cells is determined. As the CSL and MHS designs are performed sequentially and separately, the design procedure invariably leads to a solution that can be far from the total optimum [3, 4]. In recent years, researchers have focused on the concurrent design of both CSL and MHS design by adopting an integrated approach [5- 7].

Owing to computational complexity very little work has been done to solve the Integrated CSL and MHS design problem simultaneously [8,9]. In this work an attempt has been made to develop a methodology to solve a bi-criteria optimization problem

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 323–331, 2011.
© Springer-Verlag Berlin Heidelberg 2011

in an integrated layout design. The proposed algorithm is experimented with using a wide range of problem instances and found consistent in producing satisfactory solutions within a reasonable period of computational time.

2 Problem Description

There are *N* number of cells which are to be placed in a production floor layout of width *W* and height *H*. The cells are considered to be rectangular blocks of known dimensions. Given the width and height of the individual cell (determined by the size and shape of the facilities), the predefined Input and Output (I/O) stations at the boundary of the cell and quantum and frequency of material flow between the cells, the aim is to find the exact location (*x* and *y* coordinates) and orientation of the individual cells, and to decide the aisle distance between the cells (along the department perimeter) with the objective of minimizing the total material handling cost and distance weighted cost of total closeness rating score.

In any facility layout design problem, while it is imperative to minimize the total material handling cost which is directly proportional to the distance between the cells, often it also essential to place certain cells as farthest as possible depending upon the nature of the production process, the safety issues and the like. Hence these two important objectives are considered in this work. The abbreviations used in the model are:

N	the total number of cells in the layout
W	the width of the floor space
H	the height of the floor space
i,j	indices to denote cells
f_{ij}	the directed flow density from cell *i* to cell *j*
w_i	width of cell *i* in the initial orientation
h_i	height of cell *i* in the initial orientation
(I_i^x, I_i^y), (x_i^I, y_i^I)	local coordinates, spatial coordinates of the input station of cell *i*
(O_i^x, O_i^y), (x_i^O, y_i^O)	local coordinates, spatial coordinates of the output station of cell *i*
(x_i, y_i), (x_i', y_i')	spatial coordinates of the lower-left corner, the upper- right corner of cell *i*
l_{ij}	equals 1 if cell *i* is placed to the left of cell *j* (that is $x_i' \le x_j$) and 0 otherwise
b_{ij}	equals 1 if cell *i* is placed below cell *j* (that is $y_i' \le y_j$) and 0 otherwise
d_{ij}	shortest contour distance from the output station of cell *i* to the input station of cell *j*
$(u_i, v_i) =$	$\begin{cases} (0,0) \text{ cell i in its original orientation} \\ (1,0) \text{ cell i is rotated } 90^{o} \text{ clockwise from its original orientation} \\ (0,1) \text{ cell i is rotated } 180^{o} \text{ clockwise from its original orientation} \\ (1,1) \text{ cell i is rotated } 270^{o} \text{ clockwise from its original orientation} \end{cases}$
c_{ij}	the cost of travel of unit material for unit distance between cell *i* and *j*
r_{ij}	closeness rating score between cell *i* and *j*

W_1 weightage for total material handling cost

W_2 weightage for distance weighted cost of total closeness rating score ($W_2=1-W_1$)

The mathematical model for the Bi-criteria integrated layout design problem is formulated based on the model represented in [5] and shown below:

$$\text{Minimize Combined Objective Function (COF)} = \sum_{i=1}^{N}\sum_{j=1}^{N}(W_1 c_{ij} f_{ij} d_{ij} + W_2 r_{ij} d_{ij}) + P \qquad (1)$$

where $P = \alpha(P_w + P_h)$ is a penalty term to guarantee that the layout solution satisfies the following floor boundary condition:

$$x_i' \le w \ \forall i, \ \ y_i' \le H \ \forall i$$

$$P_w = \max\{0, \max_i\{x_i'\} - W\}, \quad P_h = \max\{0, \max_i\{y_i'\} - H\}$$

α = the weight for penalty and was set to be algebraic sum of flow interaction between each pair of cells.

subject to :

$$x_i' = x_i + (1-u_i)w_i + u_i h_i \qquad \forall i \qquad (2)$$

$$y_i' = y_i + (1-u_i)h_i + u_i w_i \qquad \forall i \qquad (3)$$

$$x_i^{I(O)} = x_i + (1-u_i)(1-v_i)I(O_i^x) + I(O_i^y)u_i(1-v_i) + (w_i - I(O_i^x))(1-u_i)v_i + (h_i - I(O_i^y))u_i v_i \qquad \forall i \qquad (4)$$

$$y_i^{I(O)} = y_i + (1-u_i)(1-v_i)I(O_i^y) + (w_i - I(O_i^x))u_i(1-v_i) + (h_i - I(O_i^y))(1-u_i)v_i + I(O_i^x)u_i v_i \qquad \forall i \qquad (5)$$

$$l_{ij} + l_{ji} + b_{ij} + b_{ji} \ge 1 \qquad \forall i \qquad (6)$$

$$x_i' \le l_{ij} x_j + W(1-l_{ij}) \qquad \forall i < j \qquad (7)$$

$$y_i' \le b_{ij} y_j + H(1-b_{ij}) \qquad \forall i < j \qquad (8)$$

$$x_i', y_i', x_i^{I(O)}, y_i^{I(O)} \ge 0 \qquad \forall i \qquad (9)$$

$$u_i, v_i \in \{0,1\} \qquad \forall i \qquad (10)$$

$$l_{ij}, b_{ij} \in \{0,1\} \qquad \forall i,j \qquad (11)$$

In equation (1) the first term comprises the summation of two weighted-cost components, viz., (1) $W_1 c_{ij} f_{ij} d_{ij}$ represents the total material handling cost between cell i and cell j which are separated by a distance d_{ij} involving a flow intensity f_{ij} and cost c_{ij}. (2) $W_2 r_{ij} d_{ij}$ represents the weighted closeness rating cost between cell i and cell j with a closeness rating score r_{ij}.

Constraints (2) and (3) define the x-coordinate of the right boundary and the y-coordinate of the upper boundary of each cell. Constraints (4) and (5) are used to specify the x and y coordinates of I/O stations for each cell. These coordinates are expressed in generalized terms with respect to the lower-left corner point of the cell under the horizontal configuration, that is, before considering rotation.

Constraints (6), (7) and (8) are to ensure that there is no overlap between any pair of cells by letting each pair of cells be separated in the x or y direction. Constraints (9), (10) and (11) specify the bounds for each variable.

Using the predefined closeness matrix that stipulates the required degree of closeness between the cells and the following lookup table, the closeness rating score r_{ij} between any two cells i and j is computed.

Degree of closeness	Assigned Score
Absolutely necessary	5
Especially important	4
Important	3
Ordinary closeness is okay	2
Unimportant	0
Undesirable	-5

The closeness matrices for test problems are generated at random.

3 Proposed Methodology

In this work, a simple GA is proposed to obtain the best feasible solution which minimizes simultaneously the total material handling cost and distance-weighted cost of total closeness rating score.

3.1 Solution Representation

In a GA approach feasible solutions to the problem are encoded into a string of decision choices that resemble chromosomes. The chromosome that represents a feasible solution is shown in Fig 1.

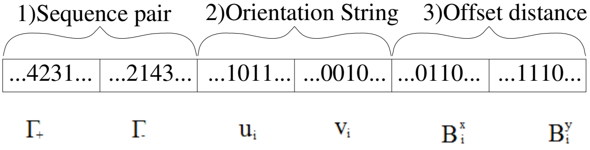


Fig. 1. Chromosome Structure

The chromosome string consists of three parts. For a layout problem of N cells, the first part is the first and second sequence (Γ_+ and Γ_-) of the sequence pair, the second part is a binary code of $2N$ bits that represents u_i and v_i of each cell, and the last part is $2N$ bytes which helps to define the offset distances in the x and y directions for each cell. The offset distances, Δx_i and Δy_i, for cell i are determined as follows: $\Delta x_i = (B_i^x/255) * \Delta X$ and $\Delta y_i = (B_i^y/255) * \Delta Y$. Where ΔX and ΔY are two preset constants and are problem dependent. In this study, they are set such that $\Delta X = \Delta Y = min\{\,\underset{i}{min}\{w_i\},\underset{i}{min}\{h_i\}\}$

3.1.1 Sequence Pair Representation

A cell system layout (CSL) can be represented by a unique Sequence Pair (SP) describing the topology of the cell placement [10]. A CSL, shown in Fig.2, consisting of three cells *a*, *b* and *c* with dimensions (10 ×5), (5×5), and (4×8) can be represented by a unique SP = (*bac* ; *abc*). This SP defines the relative positions of the cells in the CSL. Consider cells *a* and *c* in the SP: in both the sequences *a* precedes *c* and so in the CSL *a* is to the left of cell *c*. Similarly, between cells *b* and *c*, *b* precedes *c* in both the sequences and so *b* is to the left of cell *c*. Consider cells *a* and *b*: in the first sequence *b* precedes *a* and in the second *a* precedes *b* indicating there is no horizontal relationship between cell *a* and cell *b*. As in the first sequence, *b* precedes *a* and so in the CSL, the location of cell *b* is above cell *a*.

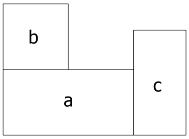


Fig. 2. CSL for SP = (*bac*; *abc*)

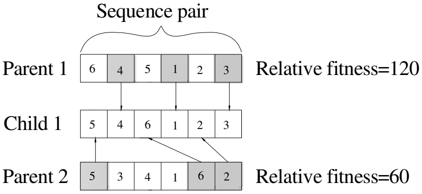


Fig. 3. Crossover operation on the first part SP = (Γ_+, Γ_-)

3.2 Fitness Evaluation

The decoding of a chromosome and finding the objective function value for a feasible solution is done in three steps.

Step 1: Using the sequence pair evaluation algorithm, Algorthim1 found in the literature [11], the spatial coordinates of the lowest left corner of each cell in a CSL is computed.

Step 2: Computing the optimal travel distance between cells (d_{ij}), through repeated applications of Dijkstra's algorithm (Cormen et al. [12]). The shortest path distance d_{ij} obtained along the department perimeter is unique for the corresponding CSL.

Step 3: Calculation of the Combined Objective Function value using equation (1).

3.3 GA Operators

3.3.1 Selection

The selection module using the Roulette wheel mechanism [16] ensures reproduction of a larger number of highly fit chromosomes compared to the number of less fit chromosomes.

3.3.2 Crossover

The crossover operation is exercised on the chromosomes of the intermediate population with a probability, known as crossover probability (p_c).The crossover operator of GA is problem dependent. For the first part, a crossover operator similar to [13] was implemented for the first and second sequence of the sequence pair independently. The first child is constructed by randomly picking a gene from the first parent and placing it in a child string at the same location as its position in the parent sequence. This process is continued for k cells, where k is proportional to the relative fitness of the first parent. The missing integers in the first child are filled in the same order as they appear in the second parent. Similarly the second child string is created by reversing the selection order of the two strings. For the last two parts of a chromosome, a heterosexual one-point crossover [14] was adopted. An example of this crossover operator is shown in Figs. 3 and 4.

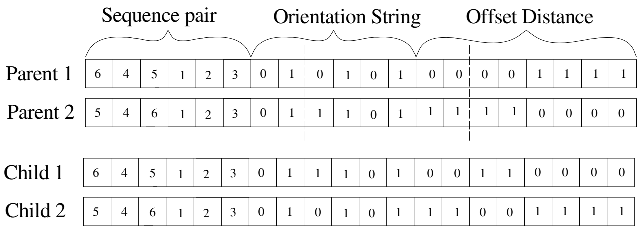


Fig. 4. Crossover operation on the second and third part

3.3.3 Mutation

The mutation operator is a mechanism that is used to divert the GA search. For the first and second sequences of the first part of the chromosome the mutation operator involves a random selection and swapping of two integers. For the second part, the mutation operator involves randomly altering one symbol with another. For the last part, the mutation operation involves replacing a randomly chosen byte with a new value generated at random in the range [0, 255]. An example of the three types of mutation operators is shown in Fig. 5.

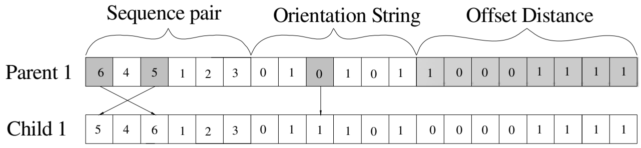


Fig. 5. Mutation operation on the first, second and third part of the chromosome

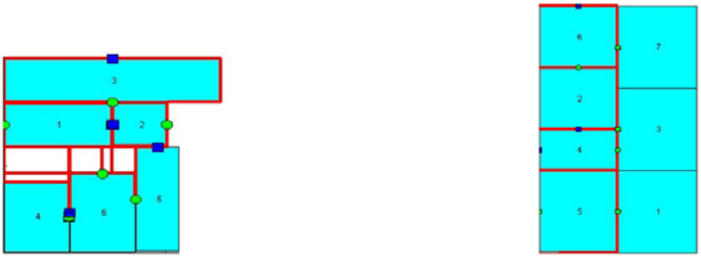
3.4 Control Parameters

The control parameter values for the GA based metaheuristic algorithm were determined based on trial experiments which produce a satisfactory output, and are summarized as below:

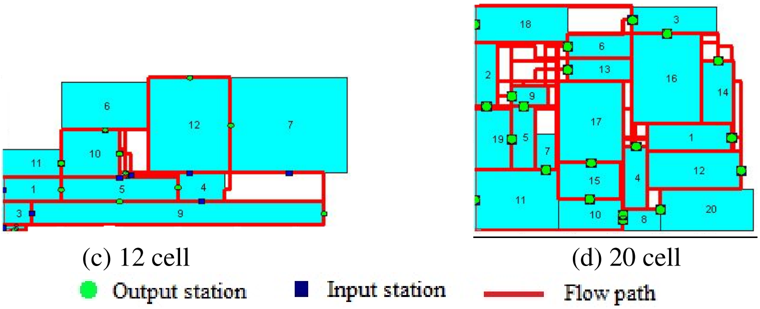
(i) Population size (P_s) = 20
(ii) Cross over probability (p_c) = 0.8
(iii) Mutation probability (p_m) = 0.25
(iv) Termination criterion (*€*): The search process is terminated if either of the following two conditions is satisfied: First, the whole process of GA based metaheuristic algorithm is terminated after '$€_1$' consecutive iterations. After many trials it was found that 1000 is the best value for '$€_1$'. Second, the search will also stop if the current best solution remains unchanged for'$€_2$' subsequent generations. After many trails, it was found that 10 is the best value for'$€_2$'.

4 Results and Discussion

The proposed genetic algorithm based procedure was coded in MATLAB and implemented in a Dual-core processor with 2GB RAM. Experiments were conducted using the bench mark problems found in the literature [6, 7, and 15]. For each bench mark problem, 10 different initial population sets, each population set having 20 different initial solutions, were generated at random. The experiment with each initial population was repeated 10 times and the best solution obtained for each of the bench mark problem is reported (Fig 6 and Table 1). The average computational time taken by the algorithm to reach the optimal solution is given in Table 2.



(a). 6 cell (b) 7 cell



(c) 12 cell (d) 20 cell

Fig. 6. The best layouts obtained by GA for the 4 test problems. a–d. W_1=0.5

Table 1. Bi-criteria values for the optimal solutions obtained

No of cells	W_1	W_2	OF_1	OF_2	Normalized OF_1	Normalized OF_2	*Best COF*	Average *COF*
6	0.5	0.5	706	254	0.841	0.678	0.76	0.967
7	0.5	0.5	182500	13900	0.652	0.772	0.712	0.799
12	0.5	0.5	9124	7003	0.797	0.778	0.787	0.984
20	0.5	0.5	1728415	352608	0.72	0.392	0.556	0.902

OF_1 – Objective Function 1 value; OF_2 - Objective Function 2 value; *COF* – Combined Objective Function

Table 2. Average Computational time (s)

No of cells	6	7	12	20
Proposed GA	425	445	2750	16800

The proposed algorithm is consistent in producing solutions, from each experiment that are closer to the best found solution.

5 Conclusion and Scope for Future Research

In this work, to overcome the limitations of the sequential design procedure, an integrated design of CSL and MHS was adopted. Again, to simultaneously optimize two different objectives of the layout design problem, a Genetic Algorithm based procedure was proposed. The proposed algorithm was tested with four different benchmark problems of different problem sizes to concurrently optimize two objectives, namely 1. Total material handling cost 2. Distance-weighted cost of closeness rating score. It was found that the proposed algorithm is able to produce satisfactory solutions consistently within an acceptable computational time limit.

The outcome of this research leaves scope for further research towards employing a local search mechanism to further reduce the computational time. To provide a wide range of alterative solutions to the implementers, a GA-based multi objective evolutionary algorithm can be developed to produce a Pareto optimal front for problems with more than two objectives.

References

1. Kim, J.G., Kim, Y.D.: A space partitioning method for facility layout problems with shape constraints. IIE Trans. 30, 947–957 (1998)
2. Tate, D.M., Smith, E.A.: Unequal-area facility by genetic search. IIE Trans. 27, 465–472 (1995)
3. Rajasekharan, M., Peters, B.A., Yang, T.: A genetic algorithm for facility layout design in flexible manufacturing systems. Int. J. Prod. Res. 36, 95–110 (1998)
4. Deb, S.K., Bhattacharyya, B.: Solution of facility layout problems with pickup/drop-off locations using random search techniques. Int. J. Prod. Res. 43, 4787–4812 (2005)

5. Hu, G.H., Chen, Y.P., Zhou, Z.D., Fang, H.C.: A genetic algorithm for the inter-cell layout and material handling system design. Int. J. Adv. Manuf. Technol. 34, 1153–1163 (2007)
6. Aiello, G., Enea, M., Galante, G.: An integrated approach to the facilities and material handling system design. Int. J. Prod. Res. 40, 4007–4017 (2002)
7. Wu, Y., Appleton, E.: The optimization of block layout and aisle structure by a genetic algorithm. Comput. Indust. Eng. 41, 371–387 (2002)
8. Ho, Y.C., Moodie, C.L.: A hybrid approach for concurrent layout design of cells and their flow paths in a tree configuration. Int. J. Prod. Res. 38, 895–928 (2000)
9. Norman, B.A., Arapoglu, R.A., Smith, A.E.: Integrated facilities design using a contour distance metric. IIE Trans. 33, 337–344 (2001)
10. Murata, H., Fujiyoshi, K., Kajitani, Y.: VLSI module placement based on rectangle-packing by the sequence-pair. IEEE Trans. CAD Integ. Circ. Syst. 15, 1518–1524 (1996)
11. Tang, X., Wong, D.F., Tian, R.: Fast Evaluation of Sequence Pair in Block Placement by Longest Common Subsequence Computation. In: Design, Automation and Test in Europe, pp. 106–111 (2000)
12. Cormen, T.H., Leiserson, C.E., Rivest, R.L.: Introduction to algorithms. McGraw-Hill and MIT Press, New York (1990)
13. Kochhar, J.S., Foster, B.T., Heragu, S.S.: HOPE: A genetic algorithm for the unequal area facility layout problem. Comput. Oper. Res. 25, 583–594 (1998)
14. Poli, R., Langdon, W.B.: Genetic programming with one-point crossover. In: Second On-line World Conference on Soft Computing in Engineering Design and Manufacturing, pp. 23–27. Springer, London (1997)
15. Welgama, P.S., Gibson, P.R.: A construction algorithm for the machine layout problem with fixed pick-up and drop-off points. Int. J. Prod. Res. 11, 2575–2590 (1993)
16. Goldberg, D.E.: Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, USA (1989)

Reconfigurable Composition of Web Services Using Belief Revision through Genetic Algorithm

Deivamani Mallayya and Baskaran Ramachandran

Anna University, Chennai – 600 025, Tamil Nadu, India
{deivamani,baaski}@cs.annauniv.edu

Abstract. Web Service Composition is development of customized services by discovering, integrating and executing relevant web services related to the service request. As the count of web services getting registered in the UDDI registry gets increased day by day, it makes dynamic selection of appropriate web services cumbersome. The proposed framework uses a set of beliefs using which, selection and composition are handled easily and effectively. Belief is a numerical quantity which varies with time based on the web service performance. The proposed approach focuses on the reconfiguration of service composition with the help of belief revision algorithm. The algorithm specifies a systematic approach towards service composition and reconfiguration.

Keywords: Web Service Composition, Belief Revision, Service Reconfiguration, Genetic Algorithm, Evolutionary algorithms.

1 Introduction

Web service is an atomic application which is developed with an intention to serve the requester. The growth of internet has resulted in more web services for the same functionality. The web services of same functionality, hosted in the internet mostly assure its stated functionality but differ in non functional properties of the web service which inherently makes the selection of web service much more difficult [1][2].

The service request after a series of procedures evolves as an abstract service. The corresponding WSDL is then passed to the execution engine to obtain a composite service. If the WSDL file is stored temporary for a specific time, say in a process registry and within a specific time span, if there is a similar request, the same information could be retrieved from the repository and proceed with composition [10] [11]. But this reuse of the composition may lead to some inconsistencies. Like, the services might get unavailable at run time or the service could not handle more than a specific number of requests at a time etc.

Service reconfiguration is one of the emerging initiatives in service composition. With a minor change in the held composition, an attempt was made to reconfigure resulting in a new and effective composition [6] [7]. The change could be addition/ deletion or reordering the flow among the selected web services.

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 332–340, 2011.
© Springer-Verlag Berlin Heidelberg 2011

Our approach dynamic (re)configuration incorporates belief as criteria in identifying the candidate web service or the held compositions from the registry. The flow of the composite service is based on the parameters and other characteristics decided in advance. In case of reconfiguration, the composition is revised appropriately based on genetic algorithm and belief revision[6].

The rest of this paper is organized as follows: Section 2 describes the overview of the Web Service Composition approaches. In Section 3 the belief revision is presented. In Section 4 we introduce composition and our proposed system architecture for service composition involving belief factor; Section 5 concentrates on the impact of belief reconfiguration in service composition. Section 6 gives the comparative study with existing approaches. Finally Section 7 closes this paper with a short Conclusion.

2 Related Work in Service Composition

Ontology mapping is a process of redesigning the structure of ontology to eliminate the inconsistencies that arise when a concept is added. Guilin Qi et al has devised an iterative revision algorithm for eliminating conflicts in ontology. A conflict set is identified and based on the postulates of the belief theory the idea is formulated to obtain a consistent ontology thereafter [3]. The algorithm designed in [4] merges two ontologies and removes the inconsistencies during merging for distributive environment.

In [5], the author specifies the importance of the change in the belief factor when there is a change in the structure of the belief bases. He adds that, the change in the belief base does not have an equal impact with all the agents and so they have proposed an approximate inference relation which provides partial solutions to the revision at any point of time.

The paper [11] discusses the importance of domain ontology and significance of updating the ontology when the application progresses with time to aid validity of the application. Once the services are identified, ranking among them is the most vital task. Most of the web services does the functionality but lacks in the non functional properties. So the non functional parameters like QoS parameters have become the most viable component in selecting web services [8].

The reliability quotient in measuring the level of belief is proposed in [9]. Their proposed algorithm makes the agent to decide by evaluating the incoming information, generate the consistent knowledge by reasoning and also avoid fraudulent information that is unreliable (those makes the knowledge base inconsistent).

In [12], genetic algorithms have been incorporated to tabu search to achieve higher performance. The paper [13] claims genetic algorithm to be the most eminent option if the attributes of selection is more generic. A chromosome represents set of abstract services and each entry has a pointer to the set of available web services to the abstract service. The paper [14] specifies that the inclusion of genetic algorithm provides effective composition along with updating of attributes in the nodes by using the context fee grammars.

3 Belief Revision

Belief revision is a process to maintain the belief library in a consistent state. Consider a set of belief which the system has as the belief set. The belief set is said to be consistent when two or more beliefs of the services does not contradict with each other. When new information needs to be added to the existing belief set, and if the addition of that information leads the system to an inconsistent state, belief revision is done [7][8].

3.1 Belief Sets

Let ‘*L*’ be the set of independent beliefs of web services. With ‘*L*’ as the initial ingredient, a belief base ‘*K*’ is designed without violating the integrity of the belief base. A belief say ‘*Ψ*’, exist in the belief base ‘*K*’ then it states that *K* is closed under logical consequences.

$$L=\{L_1,L_2,L_3.....L_n\}, K=\{\Psi_1,\Psi_2,\Psi_3,.....\Psi_n\}\ K\subseteq L$$

$$\Psi\subseteq K,\exists\ \text{‘}\Psi\text{ is consistent’}$$

On the contrary, the belief ‘*Ψ*’ which might lead the belief set to inconsistent state also could be added to the belief set with a negation operator.

$$K\ni\Psi;\forall\Psi\in K\Rightarrow\Psi\subseteq K$$

$$K\ni\neg\Psi;\forall\neg\Psi\in K\Rightarrow\Psi\not\subset K$$

The above two equation specifies the belief would be considered to revision only it does leads the existing belief set to an inconsistent state. Two beliefs are said to contradictory when they both can exist together. The contradiction is represented with $\perp$.

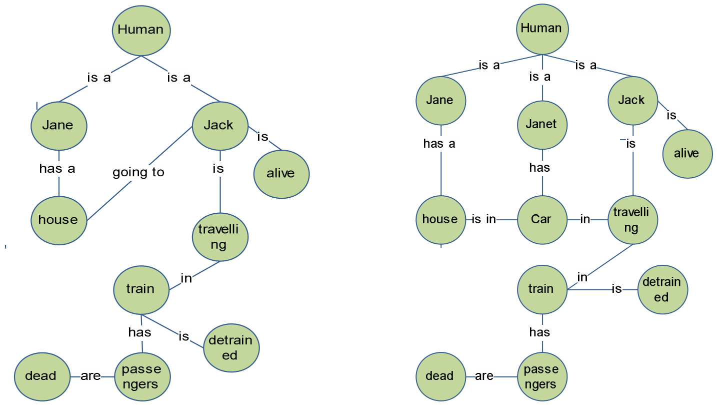


Fig. 1. Tree representation of concepts in the beliefs B1 and B2 and tree representation of the concepts in belief B1, B2 and B3

An example to arrive at a decision (conclusion) with a set of imaginary beliefs:

***B_1*:** Jack has booked a train and travelling today. ***B_2*:** The train is detrained leaving all the passengers dead. ***B_3*:** Jack has reached Jane's house in Janet's car. ***B_4*:** Jack is dead. With the 4 beliefs stated above, we arrive at the following conclusions - $C_1 \Rightarrow B_1 {}_+ B_2 = B_4$, $C_2 \Rightarrow B_4$, B_3 are contradictory, represented as B4 $\perp$ B_3. Based on the numerical value, calculated based on the relevance, credibility, semantics and other factors, the correct conclusion C_1 should be chosen. Fig 1 represents the evolution of Belief tree when addressing the beliefs.

4 Belief Revision in Service Composition

The belief is categorized as provider belief and the beneficent belief where, beneficent belief is the most important factor to be considered during composition. The rate at which the composition satisfies the service request depends on the selection of web services from the UDDI registry. This selection is based on the belief's which is calculated as given below.

4.1 Belief Calculation

Belief is calculated based on the elementary concepts discussed. We also use some more measures in determining the belief factor, which mainly derives from the below mentioned ones or could be directly specified by the beneficent. Belief factors vary with respect to provider and the beneficent. So an appropriate weight for each factor is assigned, so that effective web service could be used for composition. With the terminologies defined below, belief is defined as,

$$P_{belief} = 0.1 * WS_{avail} + 0.2 * WS_{\cos t} + 0.1 * WS_{time} + 0.3 * WS_{\sec} + 0.3 * WS_{reliable} \tag{1}$$

$$Beneficient_{belief} = 0.6 * WS_{relevance} + 0.2 * WS_{trust} + 0.2 * WS_{cre} \tag{2}$$

Table 1. Belief Factors

Belief factor	Definition	Formula Used
Relevance	Measure of correctness	$WS_{rel} = count \sum_{i=0}^{n} RC(registry, relation)$
Credibility	rate of acceptability	$WS_{cre} = \frac{Belief_{\exp ected}}{Belief_{current}}$
Availability	Time period when the service stays functional	$WS_{avail} = \frac{Service_{Uptime}}{Servic_{Uptime} + Service_{Downtime}}$
Reliability	Probability of the service to fail.	$WS_{reliable} = \frac{Time_{total} - Time_{downtime}}{Failure_{count}}$
Transaction time.	RTT of the service request and composition	$WS_{time} = (Time_{fin} - Time_{start}) + Time_{delay}$

5 Reconfiguration of Web Service Composition

The idea behind the reconfiguring the web service composition is to minimize the time to evolve a new composition. Inspired from genetic programming, composition is represented as a genome, an array comprising a set of domain concepts (or abstract web services) inferred from the service request. The various candidate services which are selected for composition after satisfying the quality and the semantic criteria as the part of pre-processing are link to the corresponding domain concepts.

The architecture is shown in Fig 2. The service request from the browser is parsed as usual after which, the process repository is searched for held compositions, represented as genomes. This could result in one of the three possibilities:

Case 1: The abstract web services of the genome satisfy the request completely.
Case 2: Some of the candidate web service pertaining to a corresponding abstract web service has to be altered.
Case 3: Some of the abstract web service needs to be altered.

Case 1 and 2 occurs when the layout of the genome needs no modification. Case 1 is the simplest of all, both the genome and candidate web services chosen are eminent for composition, so the belief tree of that composition is retrieved. A new belief tree is devised for the search request with the help of the previous belief tree.

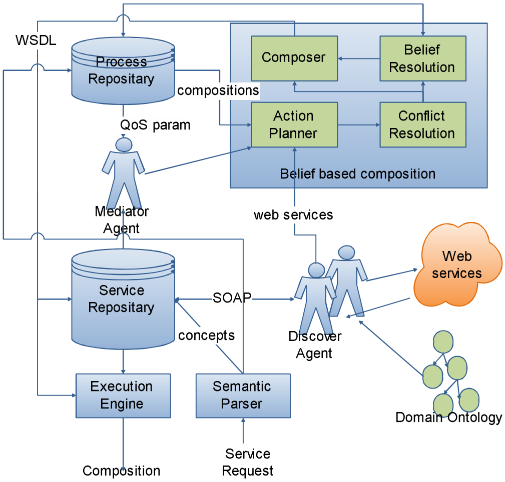


Fig. 2. WSC architecture incorporating Reconfiguration

Case 2 occurs when the base structure of the genome corresponds to the request whereas; the candidate web services needs some replacement. This process is inferred from the crossover operation in GA. This approach delivers a new composition similar to the existing genome without any alteration in the abstract web services.

Case 3 occurs if the reconfiguration demands change in the base genome itself, we have to mutate the genome. Mutating the genome implies a part of abstract web services of the genome is updated with another set of abstract web services available in another genome. If there is no other genome to mutate with, then from the existing tree of belief, a preorder traversal is done to find the node from which the current composition demands mutation. The child nodes of the identified node is detached and sent to the action planner where the composition takes place.

By following any three cases, the composition is selected and the corresponding new belief tree is fed as the input to the Action planner. The Conflict resolution engine checks for the conflicts plays a most important role as the belief tree is revised from the existing under the notion that the belief base is consistent.

If there is conflicts between the services, new services are identified with the help of discover and the mediator agent. The services along with the partial tree are fed to the Action planner to devise a control flow among the web services. When the conflict free paths are identified they are then targeted against the belief threshold to select the most eminent path and fed to the composer, which in turn creates a WSDL file specifying the flow of actions.

The main idea behind reconfiguration of web service configuration is to dynamically change the selected web services of a composition and swap the same with web services of highest belief and formalize the control flow and target the same to the WSDL file. The WSDL file is then sent to the execution engine and the procedure is followed as usual. With this approach, some amount of time in selecting the web services could be reduced.

Algorithm 1. WSC Reconfiguration (CompositionList, ServiceReq, ExpectedRes)

```
ServiceReq = $\sum_{i=1}^{n} Concept\ (i)$
DeterminedBelief=f(ServiceReq, ExpectedRes)
For each i in ServiceReq
Search(GenomeInServiceRegistry,Concept(i))
       If(Concept(i) exist as AbstractWS)
          If(checkBeleif$_{i.}$(CandidateWS) > DeterminedBelief)
               Return beliefTree;                        // CASE 1
         Else crossover(CandidateWS);                    // CASE 2
      Else mutation(concept);                            // CASE 3
BeliefTree = RetrieveBeliefTree(Concept(i))
EndFor
AddToTempCandidateWSList[]= ServiceRegistry(BeliefTree)
For each j in TempCandidateWSList
       if(checkBeleif$_i$ (WS(j) < Determined$_{Belief}$)
           s= SelectWebService(Determined$_{Belief,}$ Concept(i))
       if (s)
           AddToCandidateWSList()
       End If
End For
computeControlFlowGraph(CandidateWSList)
checkConflictsAndResolve()
checkForBeliefResolution()
configureWSDLfile(path)
```

The algorithm (algorithm 1) is used while reconfiguring the web service composition. This algorithm tries to match an existing composition to the service request by querying the process registry. It's assumed that, the WSDL files of the previous compositions along with the parsed concepts of the service request are stored in the process registry for a specific time period. The time frame is limited to deliver reliable service to the requestor.

5.1 Running Example

Our proposed belief based composition is analyzed for tourism domain. Sample ontology of tourism domain was created. As discussed, the ontology also contains the QoS information of the candidate web services. For the information related to beliefs, the process repository is accessed. Considering a query related to tourism domain, the pre-processing phases of composition have identified, train, hotel, travels and restaurant as the candidate web service. The pre-processing step also includes the computation of 'DeterminedBelief' based on the request and the expected result. Those services that satisfy DeterminedBelief' are selected our scenario resulted in 5 compositions and their respective beliefs are shown in Fig 3(a).

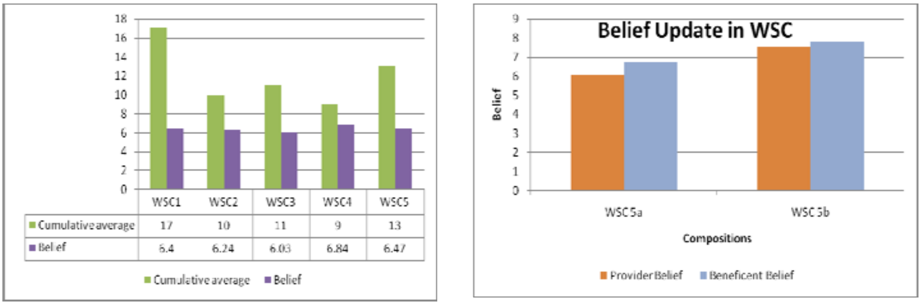


Fig. 3. (a) Experimental results of composition with Belief (b) Comparative Study

The web services WS1 through WS4 has a higher provider availed trust when compared to the actual trust factor giving by the beneficent. Comparatively, WS5 has a higher impact with respect to the beneficent than the belief factor actually quoted at the time of registering in the UDDI registry.

Suppose, after the composition is offered, subsequently if there is a service request of same type and with the required belief factor as 7.5, we could identify the best composition from the process repository for the service request available is with 6.8. When the algorithm is applied, a notable difference was identified. The composition 5 was lacking in security, reliability issues and the web service related to; 'travel' made its belief factor less comparatively. The comparison of the composition before and after the service reconfiguration is shown in Fig 3(b). In the similar way WSDL file is also altered accordingly.

6 Comparative Study

There are various approaches defined for composition in literature. Table 2 summarizes the comparative study of those approaches with the proposed approach. The comparison is based on the features like service connectivity, the strategy used for composition, inclusion of QoS, the evaluation mechanism and graph representation.

Table 2. Comparison of proposed framework with existing frameworks

Framework	Connectivity	Composition Strategy	QoS Modeling	Graph Support
BPEL4WS	Yes	Workflow	No	No
PetriNet	Yes	Workflow	Yes	Yes
Fuzzy	Yes	Workflow + AI	No	No
GA based	Yes	Workflow + AI	Yes	Yes (belief based)

7 Conclusion

The paper addresses an approach for Genetic algorithm driven belief-based service reconfiguration. The approach tries to replace each individual component of genome by using various genetic operations and evolve reconfigurations from the held compositions from the service registry as specified in the algorithm. Finally, we give an example to show how to use our approach to reconfigure composition. We believe the proposed algorithms provide a useful engineering solution to the end-to-end QoS problem for reconfiguring a SOA based software. In future, we will consider turning our method into concrete business applications.

References

1. Vengattaraman, T., Dhavachelvan, P., Baskaran, R.: An Application Perspective Evaluation of Multi-Agent System in Versatile Environments. International Journal of Expert Systems and Applications 38, 1405–1416 (2011)
2. Vengattaraman, T., Dhavachelvan, P., Ramalingam, A., Baskaran, R.: A Vibrant User Friendly Content Management System using Web Services. Accepted in the International Journal of Computer Network and Security (IJCNS) (2011) ISSN: 0975-8283
3. Qi, G., Ji, Q., Haase, P.: A Conflict-Based Operator for Mapping Revision–Theory and and Implementation. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 521–536. Springer, Heidelberg (2009)
4. Ensan, F., Bagheri, E.: A framework for handling revisions in distributed ontologies. In: SAC 2010, pp. 1417–1422 (2010)
5. Chopra, S., Parikh, R., Wasserman, R.: Approximate Belief Revision' L. Jounal of IGPL 9(6), 755–768
6. Friedman, N., Halpern, J.Y.: Belief Revision: A Critique. Presented at Journal of Logic, Language and Information, 401–420 (1999)
7. Friedman, N., Halpern, J.Y.: A knowledge-based framework for belief change. Part II: revision and update. In: Proc. Fourth International Conference on Principles of Knowledge Representation and Reasoning (KR 1994), pp. 190–201 (1994)
8. Luo, Y.-S., Qi, Y., Hou, D., Shen, L.-F., Chen, Y., Zhong, X.: A novel heuristic algorithm for QoS-aware end-to-end service composition. Science Direct (2010)
9. Huang, A.F.M., Lan, C.-W., Yang, S.J.H.: An optimal QoS-based Web service selection scheme. Information Sciences, 3309–3322 (2009)

10. Yu, T., Lin, K.-J.: Service Selection Algorithms forWeb Services with End - to- End QoS Constraints. In: Proceedings of the 2005 IEEE International Conference on e-Technology, e-Commerce and e-Service, Hong Kong, China, pp. 129–136 (2005)
11. Lv, C., Dou, W., Chen, J.: QoS-Aware Service Selection Using QDG for B2B Collaboration. In: Proceedings of the Fourteenth IEEE International Conference on Parallel and Distributed Systems, pp. 336–343 (2008)
12. Choi, S.W., Her, J.S., Kim, S.D.: QoS metrics for evaluating services from the perspective of service providers. In: Proc. IEEE Intl. Conf. on e-Business Engineering, pp. 622–625 (2007)
13. Bahadori, S., Kafi, S., Far, K.Z., Khayyambashi, M.R.: Optimal web service composition using hybrid GA-TABU search. Journal of Theoretical and Applied Information Technology, 10–15 (2009)
14. Weise, T., Bleul, S., Comes, D., Geihs, K.: Different approaches to Semantic web service composition. In: ICIW (2008)

Neural Network Based Model for Fault Diagnosis of Pneumatic Valve with Dimensionality Reduction

P. Subbaraj[1] and B. Kannapiran[2]

[1] Sri Nandhanam College of Engineering and Technology, Tirupattur-635 601,
Vellore District, Tamilnadu, India
[2] Department of Instrumentation & Control Engineering,
Arulmigu Kalasalingam college of Engineering, Anand Nagar, Krishnankoil-626190
Srivilliputhur, Virudunagar District, Tamilnadu, India

Abstract. Fault detection and diagnosis of pneumatic valve used in cooler water spray system in cement industry is of great practical significance and paramount importance for the continued operation of the plant. This paper presents the design and development of Artificial Neural Network (ANN) based model for the fault detection of pneumatic valve in cooler water spray system in cement industry. Principal component analysis (PCA) is applied to reduce the input dimension. The training and testing data required for the development of ANN is generated in a laboratory grade experimental setup. The performance of the developed model is compared with the network trained with the original variables without any dimensionality reduction. From the comparison it is observed that the classification performance of the neural network has been improved due to the application of PCA and the training time of the neural network is reduced.

Keywords: Fault diagnosis, Principal component analysis, Artificial Neural Network, Back propagation, Pneumatic valve.

1 Introduction

In cement industry the Cooler water spray system plays the important task of reducing the temperature of the clinker at the output of the kiln. The valve is one of the most important component used in cooler water spray system. There is a crucial need for checking and monitoring the condition of the pneumatic valve accurately, since they are mostly subject to hazardous environments, such as severe shocks, vibration, heat, friction, dust, etc. Defects and malfunctions (called faults) in pneumatic valve may result in significant economic loss.

Many approaches have been proposed in the literature to help maintenance engineer to do the fault diagnosis. Due to the broad scope of the process fault diagnosis problem and the difficulties in its real-time solution, many analytical-based techniques [6, 7], [11] have been proposed during the past several years for fault detection of technical plants. The important aspect of these approaches is the development of a model that describes the cause and effect relationships between the system variables

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 341–348, 2011.
© Springer-Verlag Berlin Heidelberg 2011

using state estimation or parameter estimation techniques. The problem with these mathematical model-based techniques is that under real conditions, no accurate models of the system of interest can be obtained. In that case, the better strategy is of using knowledge-based techniques where the knowledge is derived in terms of facts and rules from the description of system structure and behavior. Classical expert systems [17] were used for this purpose. The major weakness of this approach is that binary logical decisions with Boolean operators do not reflect the gradual nature of many real world problems. Also, because of the bottleneck in knowledge acquisition, the application of expert system is limited.

Later, Fuzzy Logic (FL) based techniques [5] have been proposed to develop fault diagnostic systems. The advantage of FL-based approach is that it gives possibilities to follow human's way of fault diagnosing and to handle different information and knowledge in a more efficient way. With an increasing number of variables, the possible number of rules for the system increases exponentially, this makes it difficult for experts to define a complete rule set for good system performance.

Many researchers have developed diagnostic methods by using ANNs to detect faults in plants and equipment [1, 3] [8, 9] [13, 14]. The neural network approach needs lot of data to develop the network before being put to use for real time applications. There are usually many irrelevant variables in the sampled data from real life fault diagnostic measurements. Irrelevant and redundant attributes in the input not only complicate the network structure, but also degrade the performance of the networks. By selecting only the relevant variables as input features and excluding irrelevant ones, higher performance is expected with smaller computational effort. Therefore, efficient methods for dimensionality reduction must be developed. A feasible option for solving this problem is to use feature extraction technique such as Principle Component Analysis (PCA). PCA [2] [4] [15] is a method based on the projection of the data into a low dimensional space that accurately characterises the state of the system. It produces a lower dimensional representation in a way that preserves the correlation structure between the process variables. This paper deals with the design and development artificial neural network based model for the fault detection in pneumatic valve used in cooler water spray system in cement industry, with PCA for reducing the input dimension.

The paper is organized as follows: in the next section, the description of the pneumatic valve system for this study is outlined. Section 3 describes the fault detection in pneumatic actuator. Sections 4 describe the proposed approach for fault diagnosis and section 5 describe the Dimensionality reduction using Principal Component Analysis. Sections 6 present the results and discussion and finally, in Section 7, conclusions are drawn from the work.

2 System Description

In the cement industry the output of kiln is the clinker product which is of high temperature up to 1100 °C. It is necessary to reduce the temperature of the clinker by using cooler fans and the water spray system. The description about this system is well explained in [16].

3 Fault Detection in Pneumatic Actuator

Fault detection and diagnosis are important tasks in cement industry. It deals with the timely detection, diagnosis and correction of abnormal condition of faults in the plant. Early detection and diagnosis of plants while the plant is still operating in a controllable region can help avoid abnormal event progression and reduce productivity loss. The major categories of faults in the pneumatic valve with their symbol are listed in Table 1. The sources of each fault are given in [10]. A complete description of the faults and the way they affect the valve can be found in [12].

Table 1. List of various types of faults

Types of fault	Name of the fault	Symbols	Name of the fault	Symbols
Control valve faults	valve clogging	F1	external leakage	F5
	valve seat erosion	F2	internal leakage	F6
	valve seat sedimentation	F3	critical flow	F7
	bushing friction	F4	-	-
Pneumatic servo-motor faults	twisted servo-motor's piston rod	F8	servo-motor's diaphragm perforation	F10
	terminals tightness	F9	servo-motor's spring fault	F11
Positioner faults	electro-pneumatic transducer fault (E/P)	F12	pressure sensor fault (PT)	F14
	rod displacement sensor fault (DT)	F13		
General faults/external faults	Positioner supply pressure drop	F15	partly opened bypass valves	F18
	pressure drop on valve output	F16	flow rate sensor fault (FT)	F19
	pressure drop on valve at inlet	F17	-	-

4 Proposed Approach for Fault Diagnosis

The proposed methodology for fault detection in pneumatic actuator is based on using feed forward neural network trained by back propagation to detect the abnormal conditions of the given parameters, which leads to various faults. The normal condition represents no fault situation and abnormal condition represents, fault occurrence. The main purpose of selecting ANN as a tool is due to its good generalization ability, fast real time operation and the ability to perform the complicated mapping without functional relationship. The following issues are to be addressed while developing the neural network based model for fault detection in pneumatic actuator in cooler water spray system.

4.1 Selection of Input and Output Variables

For the application of machine learning approaches, it is important to properly select the input and output variables, as ANN's are supposed to learn the relationship between input and output variables on the basis of input-output pairs provided during training. In neural network based fault detection model, the input variables represent the operating state of the pneumatic actuator, and the output is the type of fault present in the valve.

4.2 Training Data Generation

For generating the required data, various faults are introduced artificially in the experimental set up one by one and the corresponding input variables are measured. This process is repeated for various operating conditions. The pneumatic valve is made to work under different operating condition by varying the flow rate. The different faults simulated in the experimental setup for generating the data are given in Table 1.

4.3 Data Normalization

Data normalization compresses the range of training data between 0 and 1 or -1 to +1 depending on the type of transfer function. The input and output data are normalized using the expression,

$$X_n = {(X - X_{\min})}\big/{(X_{\max} - X_{\min})} + \mathit{starting\ value} \tag{1}$$

Where X_n is the normalized value of the data and X_{min} and X_{max} are the minimum and maximum values among all the values of the data.

5 Dimensionality Reduction Using Principal Component Analysis

The feature extraction technique used in this study is Principal Component Analysis (PCA). In PCA, we seek to represent the d-dimensional data in a lower-dimensional space. This will reduce the degrees of freedom; reduce the space and time complexities. The objective is to represent data in a space that best expresses the variation in a sum-squared error sense. In PCA first, the d-dimensional mean vector μ and $d \times d$ covariance matrix $\sum$ are computed for the full data set. Next, the eigenvectors and eigenvalues are computed, and sorted according to decreasing eigenvalue. Call these eigenvectors e_1 with eigenvalue λ_1, e_2 with eigenvalue λ_2, and so on. Sub-sequently, the largest k such eigenvectors are chosen. Form a k ×k matrix A whose columns consist of the k eigenvectors.

6 Results and Discussion

This section presents the details of neural network model developed for fault detection in Pneumatic valve used in cooler water spray system. Using a laboratory grade experimental setup of pneumatic valve, the required data was generated. The data contains 11 input features, which are given in Table 2 and one output that is labelled as either normal or as a fault, with exactly one specific fault. All the input features are continuous variables and the output is the fault label. The photograph of the experimental setup is shown in Figure 1.

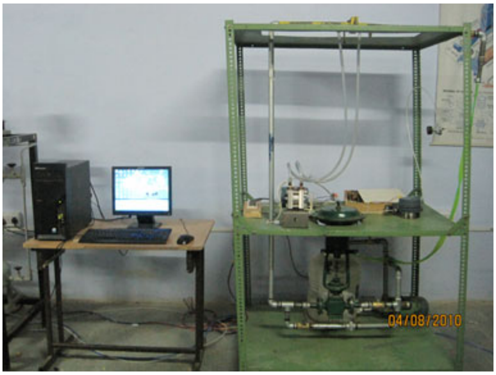


Fig. 1. Shows photograph of the pneumatic actuator system interfaced with PC using DAQ

A total of 4000 data was generated for developing the neural network, with 1000 data under normal condition and the remaining under various fault condition. The data was randomly divided into two sets, with 3000 for training and the remaining 1000 for testing. PCA was applied on the input data and the transformed components are given as input to the neural network. A feed forward neural network with 11 inputs and two outputs was used to capture the input-output relationship. Trial and error approach was followed to identify the suitable number of hidden layer neurons. The MLFFNN model used here has two hidden layers of logarithmicsigmoidal neurons. The neural network model was trained using back propagation algorithm, which propagates the error from the output layer to the hidden layer to update the weight matrix. The training function used was scaled conjugate gradient back propagation method. The mode of training used here is batch type. The performance of the ANN during training is shown in Figure 2. After training, the generalization performance of the network is evaluated with the 1000 test data that contain the combination of both normal as well as various fault categories. The trained neural network classified 1000 data correctly, which shows an overall detection rate of 100%. The network is trained until it reaches the mean square error of 0.001. The mean square error achieved during training is 0.0040. During testing, the mean square error achieved by the

Table 2. Name of the input features in the experimental system

Variable number	Name of the variable	Variable number	Name of the variable
1	Differential Pressure Transmitter output (mA)	7	Outlet pressure from the pneumatic valve
2	Temperature of water form RTD output (outlet)	8	Air inlet at the diaphragm
3	Pressure gauge reading(Inlet Pressure)	9	Water level in the tank
4	Potentiometer(Stem movement)	10	Inlet flow of the water level measuring tank
5	Input and output power of the pump	11	Speed of the pump(rpm)
6	Inlet pressure to pneumatic valve	-	-

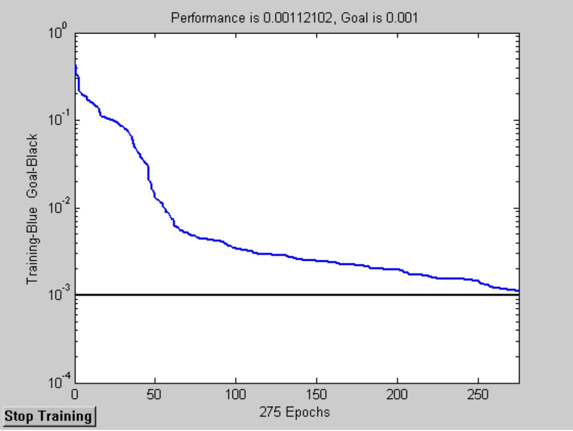


Fig. 2. Training Performance of the Network

network is 0.0041. With 9×20 hidden nodes, the network took 24.26 s to reach the error goal. This is shown in Table 3 it is found that the network has correctly classified all the data during the testing stage.

This shows that the trained ANN with PCA is able to produce the correct output even for the unseen input. Next, for comparison, the ANN model was trained with all the 11 input variables without taking the PCA and the performance of the neural network in this case is summarised in the Table 4. On comparing the performance of without dimensionality reduction, it is found that the mean square error in testing and training of using PCA is incredibly low when compared with ANN and the accuracy of fault classification is high in using PCA when compared with ANN.

The classification rate using the original parameters is 93.8%. The classification rate with feature extraction through PCA is 100%. Also, the obtained results proved that the network with PCA technique performed better than ANN without PCA. The reduced performance of the classifier without feature extraction is due to the existence of irrelevant and redundant feature parameters existing in the data set. The irrelevant

Table 3. Performance of Neural Network Model with PCA

Description of parameters	values
Number of training patterns	3000
Number of testing patterns	1000
Number of hidden nodes	9×20
Training Time	24.26 seconds
Mean Square Error in Training	0.0040
Mean Square Error in Testing	0.0041
Percentage of fault classification (Testing)	100%

Table 4. Performance comparison of ANN model with and without PCA

Description of parameters	Without dimensionality reduction	With dimensionality reduction
MSE in Training	0.0705	0.0040
MSE in Testing	0.0706	0.0041
Training Time	70.0310 seconds	24.26
Percentage of fault classified (Testing)	93.8 %	100%

features burden the classifier and tend to decrease its performance. By using PCA the useful features are extracted from the original input variables.

7 Conclusion

This paper has presented a neural network based approach with principal component analysis based feature extraction for fault diagnosis of pneumatic actuator used in cooler water spray system in cement industry. The data required for the development of neural network model have been obtained through the laboratory grade experimental setup. Totally 19 faults from the pneumatic actuator were considered in the developed model. A key issue in neural network based approach is identifying a representative set of features from which to develop the network for a particular task. This problem has been addressed by feature extraction through principal component analysis. Simulation results show that the PCA based approach is very much effective in identifying the important key features of the network. Based on the results obtained, it is observed that the performance of the neural network model has been significantly improved by reducing the input dimension. The effectiveness of the proposed method has been demonstrated through different fault condition in pneumatic actuator system.

References

1. Bayir, S.R., Bay, O.F.: Serial wound starter motor faults diagnosis using artificial neural network. In: IEEE ICM 2004, Istanbul, pp. 194–199 (2004)
2. Chang, Y.-W., Wang, Y.-C., Liu, T., Wang, Z.-J.: Fault diagnosis of a mine hoist using PCA and SVM techniques. Int. J. China Univ. Mining and Technology 18, 327–331 (2008)

3. Demetgul, M.: Experiment based fault diagnosis on bottle filling plant with LVQ artificial neural network algorithm. Journal of Engineering Science 14, 92–100 (2008)
4. de Moura, E.P., Souto, C.R., Silva, A.A., Irmao, M.A.S.: Evaluation of principal component analysis and neural network performance for bearing fault diagnosis from vibration signal processed by RS and DF analyses. Journal of Mech. Systems and Signal Processing (2010)
5. Devaraj, D., Murthy, N., Yegnanarayana, B.: A fuzzy system model for plant condition monitoring. In: ASME International Conference, pp. 210–214 (1999)
6. Isermann, R.: Supervision, fault detection and fault diagnosis methods-an introduction. Int. J. Control Engineering Practice 5(5), 639–652 (1997)
7. Isermann, R., Balle, P.: Trends in the application of Model-based fault detection and diagnosis of technical process. Int. J. Control Engineering Practice 5(5), 709–719 (1997)
8. Karpenko, M., Sepehri, N.: Neural network classifiers applied to condition monitoring of a pneumatic process valve actuator. Engineering Applications of Artificial Intelligence 15(3-4), 273–283 (2002)
9. Karpenko, M., Sepehri, N., Scuse, D.: Diagnosis of process valve actuator faults using a multilayer neural network. Control Engineering Practice 11(11), 1289–1299 (2003)
10. Koj, J.: The Fault Sources of Pneumatic Servo-Motor-Control Valve Assembly. In: Proceedings of the III Polish Nat. Conf. on Diagnosis of Industrial Processes, Jurata, Poland, pp. 415–419 (1998)
11. Leonhardt, S., Ayoubi, M.: Methods of Fault diagnosis. Int. J. Control Engineering Practice 5(5), 683–692 (1997)
12. Louro, R.: Fault Diagnosis of an Industrial Actuator Valve. MSc Dissertation, Technical University of Lisbon, Portugal (2003)
13. Rajakarunakaran, S., Venkumar, P., Devaraj, D., Rao, K.S.P.: Artificial neural network approach for fault detection in rotary system. Applied Soft Computing 8, 740–748 (2008)
14. Seong, S.H., Hur, S., Kim, J.S., Kim, J.T., Park, W.M., Lee, U.C.: Development of diagnosis algorithm for the check valve with spectral estimations and neural network models using acoustic signals. Annals of Nuclear Energy 32, 479–492 (2005)
15. Bouhouche, S., Yahi, M., Bast, J.: Combined use of principal component analysis and Self organisation map for condition monitoring in pickling process. Applied Soft Computing 11, 3075–3082 (2011)
16. Subbaraj, P., Kannapiran, B.: Artificial Neural Network Approach for Fault Detection in Pneumatic Valve in Cooler Water Spray System. Int. Journal of Comp. Applications 9(7), 975–8887 (2010)
17. Venkatasubramanian, V.: A Review of Process Fault Detection and Diagnosis Part I, II & III. Computers and Chemical Engineering (2003)

A CAD System for Breast Cancer Diagnosis Using Modified Genetic Algorithm Optimized Artificial Neural Network

J. Dheeba[1] and S. Tamil Selvi[2]

[1] Department of Computer Science and Engineering,
Noorul Islam University, Kumaracoil, TN
deeps_3u4@yahoo.com
[2] Department of Electronics and Communication Engineering,
National Engineering College, Kovilpatti, TN
tamilgopal2004@yahoo.co.in

Abstract. In this paper, a computerized scheme for automatic detection of cancerous tumors in mammograms has been examined. Diagnosis of breast tumors at the early stage is a very difficult task as the cancerous tumors are embedded in normal breast tissue structures. This paper proposes a supervised machine learning algorithm – Modified Genetic Algorithm (MGA) tuned Artificial Neural Network for detection of tumors in mammograms. Genetic Algorithm is a population based optimization algorithm based on the principle of natural evolution. By utilizing the MGA, the parameters of the Artificial Neural Network (ANN) are optimized. To increase the detection accuracy a feature extraction methodology is used to extract the texture features of the cancerous tissues and normal tissues prior to classification. Then Modified Genetic Algorithm (MGA) tuned Artificial Neural Network classifier is applied at the end to determine whether a given input data is suspicious for tumor or not. The performance of our computerized scheme is evaluated using a database of 322 mammograms originated from MIAS databases. The result shows that the proposed algorithm has a recognition score of 97.8%.

Keywords: Microcalcification, Mammograms, Computer Aided Detection, Neural Network, Texture Energy Measures, Genetic Algorithm.

1 Introduction

Breast cancer is the common disease among women in most parts on the world today. Screening mammography is one of the well known methods to detect breast cancer in woman at the early stage. Mammography reduces the mortality rate among women. The size of the tumor, age of the affected women, shape and stage predicts the survival rate. Early detection of breast cancer will increase the survival rate by 98%. Radiologists can predict signs of abnormalities with the help of visual clues in mammograms. But these visual signs are delicate and in many cases the cancer tissues are masked by normal breast tissues and make detection a challenging task.

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 349–357, 2011.
© Springer-Verlag Berlin Heidelberg 2011

Computer Aided Detection (CAD) [7,8,10] system can help the radiologists in analyzing the mammograms in an efficient way. CAD systems are used for classification of abnormal and normal tissues in the breast with the help of certain features obtained from both the tissues (abnormal and normal). Breast cancer can be seen as calcifications or as masses. Microcalcifications (MC) are quiet tiny bits of calcium, and may show up in clusters or in patterns and are associated with extra cell activity in breast tissue. Scattered MC can indicate early breast cancer, which are usually a sign of benign breast cancer. MC in the breast show up as white speckles on breast x-rays. The calcifications are small; usually varying from 100 micrometer to 300 micrometer, but in reality may be as large as 2 mm. Tumor masses in the breast are breast tissues that is growing in an abnormal and uncontrolled way.

According to American Cancer Society, some studies have shown that CAD can help find cancers that radiologist otherwise might have missed. A number of Computer Aided Detection schemes have been adopted for the detection of MC clusters in mammograms. The CAD system will take a small ROI as a subject of recognition. The CAD system will also assists the radiologists towards potential abnormalities. Research in the field of applying intelligent algorithms for classification of tumor masses and calcifications have been conducted in last few years.

Yu and Ling [1] have proposed a CAD system automatic detection of clustered microcalcifications in digitized mammograms. Using mixed features the potential microcalcification clusters are segmented, then the individual microcalcification cluster is segmented using 31 feature. The discriminatory power of these features is analyzed using general regression neural networks via sequential forward and sequential backward selection methods. Netsch and Heinz-Otto [2], uses the Laplacian scale-space representation of the mammogram. First, possible locations of microcalcifications are identified as local maxima in the filtered image on a range of scales. For each finding, the size and local contrast is estimated, based on the Laplacian response denoted as the scale-space signature. A finding is marked as a microcalcification if the estimated contrast is larger than a predefined threshold which depends on the size of the finding.

Berman Sahiner et al. used a Convolution Neural Network (CNN) classifier to classifier the masses and the normal breast tissue [3]. First, the Region of Interest (ROI) of the image is taken and it was subjected to averaging and subsampling. Second, gray level difference statistics (GLDS) and spatial gray level dependence (SGLD) features were computed from different subregions. The computed features were given as input to the CNN classifier.

Cascio.D [4] developed an automatic CAD scheme for mammographic interpretations. The scheme makes use of the Artificial Neural Network to classify mass lesions using the geometric information and shape parameters as input to the classifier. Jong and Hyyun [5] proposed a three layer Backpropagation Neural Network (BPNN) for automatic detection of microcalcification clusters. Texture features are extracted to classify the ROI containing clustered MC and ROI containing normal tissues. Chen et al. [6] presents a new texture shape feature coding (TSFS) based classification method for classifying masses on mammograms. A texture shape histogram is used for generating various shape features of masses. Oliver et al, [9] presents a comparative study of various mass detection methods and also analyzes their advantages and disadvantages.

Texture analysis is an active field of research and large number of schemes for texture feature extraction has been proposed [13]. Christodoulos et al, [11] surveyed and compared multiple texture features for classification purpose in terms of accuracy and processing time and showed that Laws Texture Energy Measures (LTEM) takes less processing time. Anna et al. [12] suggests that LTEM has a best feature in analyzing texture of tissue for BC diagnosis and achieved 90% accuracy.

A proper pattern classifier has to be employed to identify the true microcalcification clusters. Classifiers like Artificial Neural Network (ANN) are powerful tool for pattern recognition as they have the ability to learn complex, nonlinear surfaces among different classes. Brijesh and John [14] has analyzed the classification performance by varying the hidden neurons, momentum and learning rate and showed that there is an increase in classification accuracy when there is a proper parameter setting. Though ANN perform well, they have certain drawbacks due to the specific structure i.e. the back propagation (BP) algorithm employed to train the ANN will require the parameters (hidden neurons, learning rate, momentum factor) to be set manually. Moreover BP algorithms get trapped into a local minimum, which makes the algorithm entirely dependent on the initial parameter settings. To overcome these deficiencies, in this paper we propose a Modified Genetic Algorithm (MGA), which optimally designs the ANN for tumor detection. Parameters like the hidden neuron, learning rate, momentum factor are optimally selected thereby adjusting the connection weights. Genetic algorithms can be used to solve discrete optimization problems by searching the solution space [17]. In order to increase the computational efficiency of GA, an MGA is followed in this paper.

2 Methods

2.1 Mammographic Data Preprocessing

The goal of preprocessing the image is to simplify recognition of tumors without throwing away any important information. Mammograms has breast region and is superimposed over complex background structures to which analysis is not necessary. One way would be to restrict the analysis to Region of Interest (ROIs) that do not contain any background. Preprocessing is done to the mammograms to remove normal background structures. The preprocessed image is considered as the ROI image, which is subjected to feature extraction phase. The proposed CAD mechanism is shown in Fig 1.

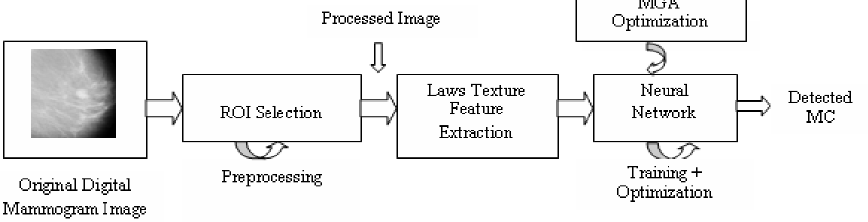


Fig. 1. Block Diagram of the Proposed CAD System

2.2 Texture Feature Extraction

In the feature extraction stage, texture features were extracted from the ROI containing MC and also from the ROI that does not contain MC. Texture contains important information, which is used by humans for the interpretation and the analysis of many types of images. Texture refers to the spatial interrelationships and arrangement of the basic elements of an image. Visually, these spatial interrelationships and arrangements of the image pixels are seen as variations in the intensity patterns or gray tones. Therefore, texture features have to be derived from the gray tones of the image. The texture energy measures developed by Kenneth Ivan Laws at the University of Southern California have been used for many diverse applications [16]. A 2-D convolution kernel for texture discrimination is generated from the following set of 1-D convolution kernels of length five. The texture descriptions used are level (L), edge (E), spot(S), wave (W) and ripple (R). L5 = (1, 4, 6, 4, 1), E5 = (-1, -2, 0, 2, 1), S5 = (-1, 0, 2, 0, -1), W5 = (-1, 2, 0, -2, 1), R5 = (1, -4, 6, -4, 1).

From this above 1-D convolution kernels 25 different two dimensional convolution kernels (masks) are generated by convoluting a vertical 1-D kernel with a horizontal 1-D kernel. To generate the Texture Energy Measures (TEM) at each pixel, these masks are convolved with the image pixel.

2.3 Modified Genetic Algorithm

Genetic algorithms are a general purpose search algorithm that uses principles from natural evolution. GA starts randomly by selecting an initial population of likely problem solutions. At each generation, each element of the current solution space (chromosomes) is evaluated and assigned a measure of its fitness. Based on the fitness value the chromosomes are selected for further genetic operations – selection, crossover and mutation. In modified genetic Algorithm an improved evolutionary direction operator and also a modified mutation scheme is introduced which improves the solution search and also increase the solution diversity.

Selection: Select an integer N_P for population size and determine the problem space according the prior knowledge. The MGA employs the improved evolutionary direction operator (IEDO) [15] for selection of chromosomes. The IEDO chooses three best solutions in each generation and then obtain a new solution that is better than original best solution. A decision variable x_i is represented by a real number and is limited to lower and upper limit ($x_i^{\min}$, $x_i^{\max}$) i.e. $x_i \in (x_i^{\min}, x_i^{\max})$. Three sets of optimal chromosomes are obtained after each generation and these three chromosomes are selected based on the best fitness value. Fitness is the only information used to assess the performance of the individual members of a population relative to the rest of the population. The fitness is defined in equation (4). Since this selection process enhances the local searching process this will reduce the effort in searching the optimal solution.

Crossover: The three preferred individuals generated in the selection operation are used for crossover operation. This operation creates new individuals by combining parts from the three individuals. The process involves uniform random selection of crossover points between the three individuals. 35% of the individuals are selected from the first preferred individual, 25% of the individuals are selected from the second preferred individual and 15% of the individuals are selected from the third preferred individual. The remaining 25% of the population are randomly selected.

Mutation: It is the process which creates new individuals by making changes in a single individual after the crossover operation. The mutation operation follows the migration operation [15] to generate a newly diverse population, which increases the search space. The nonuniform mutation operator is employed here. The new value x_i^G is regenerated as,

$$x_i^{G+1} = \begin{cases} x_i^{G+1} + \rho\,(x_i^{\min} - x_i^G)\,, \mathit{if} & \tau < \dfrac{x_i^{G+1} - x_i^{\min}}{x_i^{\max} - x_i^{\min}} \\ x_i^{G+1} + \rho\,(x_i^{\max} - x_i^G)\,, & \mathit{otherwise} \end{cases} \tag{1}$$

Where $i = 1,2...n$ and n the total number of individuals in a population N_P, τ and ρ are random numbers in the range [0, 1]

```
randomly generate initial population
    for $i = 1$ to $population\_size$
            Calculate fitness value using equation (4)
    End
        While NOT termination condition do
          1. Select three Chromosomes using the IEDO operations
          2. Perform crossover operation over the individuals
          3. Perform improved mutation scheme using equation(1)
    for $i = 1$ to $population\_size$
            Calculate fitness value using equation (4)
    End
            Choose the best solution based on fitness
        End while
```

3 Classification Using MGA Tuned ANN

The mammographic images used in this study originate from the MIAS database digital mammograms [16] generated by UK research department. The database contains left and right breast images of 161 patients. Its quantity consists of 322 images, which belongs to three types such as Normal, benign and malignant. The database has been reduced to 200 micron pixel edge, so that all images are 1024 x 1024. There are 208 normal, 63 benign and 51 malignant (abnormal) images. The 322 mammographic images were manually classified by the radiologists into four different types of abnormalities: architectural distortions, stellate lesions, Circumscribed masses and calcifications.

In the proposed Modified Genetic Algorithm tuned Artificial Neural Network (MGAANN) method, ANN is applied for evolving fully connected feedforward Neural Network and is optimized with best network architecture by optimizing the number of neurons in the hidden layer, the learning rate and the momentum factor. Finding an optimal learning rate avoids major disruption of the direction of learning when very unusual pair of training patterns is presented. The main advantage of using optimal momentum factor is to accelerate the convergence of error propagation algorithm. The number of neurons in the input layer and output layer is fixed based on the problem defined. Let N_I represents the size of the neurons in the input layer and N_O represents the size of the neurons in the output layer. The number of neurons in the input and output layer are fixed and they are same for the entire configuration in the architecture spaces. The number of hidden layers in this problem is restricted and made as one. The range of the optimization process is defined by two range arrays $R_{\min} = \{Nh_{\min}, Lr_{\min}, Mc_{\min}\}$ and $R_{\max} = \{Nh_{\max}, Lr_{\max}, Mc_{\max}\}$ where, Nh is the number of neurons in the hidden layer, Lr is the learning rate and Mc is the momentum factor. Let f be the activation function and is defined as the sum of the weighted inputs plus the bias and is represented as,

$$y_k^p = f(s_k^p) \tag{2}$$

Where $s_k^p = \sum_j w_{j,k} y_j^p + \theta_k$, y_k^p is the output of the k^{th} neuron when a pattern p is fed, $w_{j,k}$ is the weight from the j^{th} neuron and θ_k is the bias value of the k^{th} neuron in the hidden layer and it is defined by hyperbolic tangent activation function,

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{3}$$

The fitness function sought for optimal training is the Mean Square Error (MSE) formulated as,

$$MSE = \sum_{p \in T} \sum_{k=1}^{N_O} (t_k^p - y_k^{p,o})^2 \tag{4}$$

Where t_k^p is the target (desired) output, $y_k^{p,o}$ is the actual output from the k^{th} neuron in the output layer o, for the pattern p in the training set. With the framed fitness function the MGAANN algorithm automatically evolve a best solution.

The algorithm has been designed in a framework of MATLAB 7.10, which aims at developing a CAD system for breast cancer detection. The optimally designed ANN has three-layer architecture: an input layer, hidden layer and an output layer. The number of neurons that structures the input layer is equal to the number of feature vectors extracted (25 TEM). The hidden layer neurons are optimally added to the

ANN and are defined by the hyperbolic tangent activation function as in equation (4). The output layer contains one neuron which discriminates presence of MC cluster. The neural network architecture space is defined over a multilayer perceptron with the parameters $R_{\min} = \{Nh_{\min}, Lr_{\min}, Mc_{\min}\}$ and $R_{\max} = \{Nh_{\max}, Lr_{\max}, Mc_{\max}\}$.

For the classification experiments, the training samples in the features extraction phase are created by using 50 ROI's. The individual tumor pixels are obtained from 25 ROI's (12 benign and 13 malignant cases). The abnormal areas are provided by the truth markings in the mammogram image of MIAS database. The non tumor pixels are obtained from the remaining 25 ROI's. The training dataset contain a total of 2050 TEM patterns from the MIAS database. These patterns contains TEM pixels including true individual microcalcification clusters, circumscribed masses, ill defined masses and also pixels indicating normal tissues that includes blood vessels and dense breast tissues. The optimization of MGAANN classifier is performed with the learning rate and the momentum constant varied from 0 to 1 and the hidden neurons varied from 31 to 200. The crossover probability of the MGA is chosen as 0.8 and the mutation probability is chosen as 0.15. For this training a maximum of 500 generations are performed with a population size of 50 and with 500 training epochs. During each MGA generation, the best fitness score (minimum MSE) achieved by the population at the optimum dimension is stored. Using the proposed algorithm an optimized ANN is achieved with Nh =132, Lr = 0.00127 and Mc = 0.9041. Thus, the proposed algorithm yields a compact network configuration in the architecture space rather than the complex ones as long as optimality prevails. Testing is done to all images in the database.

Fig 2 shows the detection results for identifying tumor masses in mammograms. Fig 2 (a) shows the original mammogram image with ill defined masses present in it. Fig 2 (b) shows the result after the preprocessing procedure, i.e. the ROI area. Fig 2 (c) shows the final output of the MGAANN classifier where the tumor mass region is marked using red circles.

Fig.3 shows the detection results for identifying microcalcifications in mammograms. Fig 3 (a) shows the original mammogram image with microcalcification clusters present in it. Fig 3 (b) shows the result after the preprocessing procedure, i.e. the ROI area. Fig 3 (c) shows the final output of the MGAANN classifier where the abnormal region is marked using red circles. The performance of MGAANN classifier is compared with the other classifier [18] and it shows a better results.

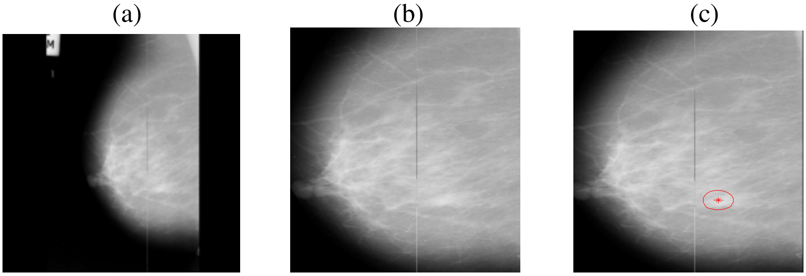


Fig. 2. Classification Results for ill defined tumor masses (a) Original Image (mdb013) (b) ROI Image (c) MGAANN Classifier output with classified masses

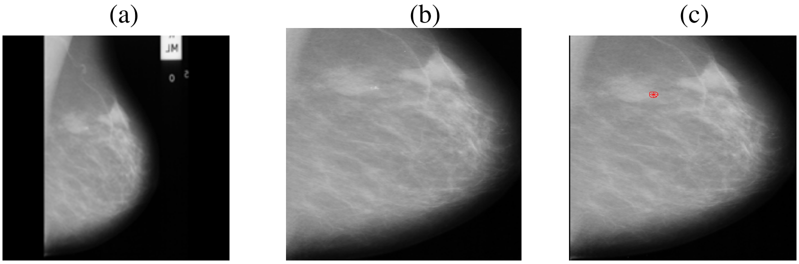


Fig. 3. Classification Results for Microcalcification (a) Original Image (mdb248) (b) ROI Image (c) MGAANN Classifier output with classified microcalcification

4 Conclusion

In this paper a novel method is followed for classification of cancer in digital mammograms. The MGAANN detects the tumor masses and microcalcification clusters in an efficient way. The algorithm developed here classifies mammograms into normal & abnormal. First, the mammogram image is enhanced using the thickening process. The Laws texture features are extracted and classification approaches using MGAANN shows good classification results. Using the mammographic data from the Mammographic Image Analysis Society (MIAS) database a recognition score of 97.8% was achieved using the proposed approach. The inclusion of the IEDO operator in the MGA algorithm helps to efficiently search and explore the solution quickly. Moreover, the proposed approach has the following merits: easy to implement; better effectiveness than previous methods; robustness of algorithm; applicable to the larger-scale system dealing with images.

References

1. Yu, S., Guan, L.: A CAD System for the Automatic Detection of Clustered Microcalcifications in Digitized Mammogram Films. IEEE Transactions on Medical Imaging 19(2), 115–126 (2000)
2. Netsch, T., Peitgen, H.O.: Scale-Space Signatures for the Detection of Clustered Microcalcifications in Digital Mammograms. IEEE Transactions on Medical Imaging 18(9), 774–786 (1999)
3. Sahiner, B., et al.: Classificaiton of Mass and Normal Breast Tissue: A convolution Neural Network classifier with spatial domain and Texture Images. IEEE Trans. on Medical Imaging 15(5), 598–609 (1996)
4. Cascio, D., et al.: Mammogram Segmentation by Contour Searching and Mass Lesions Classification with Neural Network. IEEE Trans. on Nuclear Science 53(5), 2827–2833 (2006)
5. Kim, J.K., Park, H.W.: Statistical Textural Features for Detection of Microcalcifications in Digitized Mammograms. IEEE Trans. on Medical Imaging 18(3), 231–238 (1999)

6. Chen, Y., Chang, C.: New Texture shape feature coding based computer aided diagnostic methods for classification of masses on mammograms. In: 26th IEEE Annual Int. Conference of the Engg. In Medicine and Biology Society IEMBS, vol. 1, pp. 1275–1281 (2004)
7. Gao, X., Wang, Y., Li, X., Tao, D.: On Combining Morphological Component Analysis and Concentric Morphology Model for Mammographic Mass Detection. IEEE Trans. on Information Technology in Biomedicine 14(2), 266–273 (2010)
8. Nishikawa, R.M., Giger, M.L., Doi, K., Vyborny, C.J., Schmidt, R.A.: Computer-Aided Detection of Clustered Microcalcifications on Digital Mammograms. Medical and Biological Engineering and Computing 33(2), 174–178 (1995)
9. Oliver, A., Freixenet, J., Marti, J., Perez, E., Pont, J., Denton, E.R.E., Zwiggelaar, R.: A Review of Automatic Mass Detection and Segmentation in Mammographic Images. Journal of Medical Image Analysis 14, 87–110 (2010)
10. Wang, D., Shi, L., Heng, P.A.: Automatic Detection of Breast Cancers in Mammograms Using Structured Support Vector Machines. Journal of Neurocomputing 72, 3296–3302 (2009)
11. Christodoulou, C.I., Michaelides, S.C., Pattichis, C.S.: Multifeature Texture Analysis for the Classification of Clouds in Satellite Imagery. IEEE Transactions on Geoscience And Remote Sensing 41(11), 2662–2668 (2003)
12. Anna, N., Ioannis, S., Spyros, G., Filippos, N.: Breast Cancer Diagnosis: Analyzing Texture of Tissue Surrounding Microcalcifications. IEEE Trans. on Information Technology in Biomedicine 12(6), 731–738 (2008)
13. Reed, R.T., du Buf, J.M.H.: A review of recent texture segmentation and feature extraction techniques. Comput. Vis. Graphics Image Processing 57(3), 359–372 (1993)
14. Verma, B., Zakos, J.: A Computer-Aided Diagnosis System for Digital Mammograms Based on Fuzzy-Neural and Feature Extraction Techniques. IEEE Transactions on Information Technology in Biomedicine 5(1), 46–54 (2001)
15. Chiang, C.-L.: Improved Genetic Algorithm for Power Economic Dispatch of Units with Valve-Point Effects and Multiple Fuels. IEEE Transactions on Power Systems 20(4), 1690–1699 (2005)
16. Suckling, J., Parker, J.: The Mammographic Images Analysis Society Digital Mammogram Database. In: Proc. of 2nd Int. Workshop Digital Mammography, U.K, pp. 375–378 (1994)
17. Rajasekaran, S., Vijayalakshmi Pai, G.A.: Neural Networks, Fuzzy Logic and Genetic Algorithms. Prentice Hall of India (2000)
18. Dheeba, J, Tamil Selvi, S.: Screening Mammogram Images for Abnormalities using Radial Basis Function Neural Network. In: IEEE International Conference- ICCCCT 2010, pp. 554–559 (2010)

Application of ANN Based Pattern Recognition Technique for the Protection of 3-Phase Power Transformer

Harish Balaga[1,*], D.N. Vishwakarma[1], and Amrita Sinha[2]

[1] Dept. of Electrical Engineering, Institute of Technology, Banaras Hindu University,
Varanasi, UP, India
{harish.balaga.rs.eee,dnv.eee}@itbhu.ac.in
[2] Dept. of Electrical Engineering, NIT Patna, Patna, Bihar, India
amrita_bhu@yahoo.co.in

Abstract. This paper presents an application of ANN based Pattern Recognition Technique for the differential protection of a two winding three-phase power transformer. It proposes a variation in feed forward back propagation neural network (FFBPNN) model, which makes the discrimination among normal, magnetizing inrush, over-excitation and internal fault currents efficiently. Fault conditions of the transformer are simulated using MATLAB/SIMULINK in order to obtain current signals. The training process for the neural network and fault diagnosis decision are implemented using toolboxes on MATLAB.

Keywords: Artificial neural network, differential protection, Pattern Recognition, Power transformer.

1 Introduction

Now-a-days, research on artificial intelligence and its applications has been in full swing, particularly in the field of pattern recognition [1, 4, 15]. The ANN-based algorithms have been successfully implemented in many pattern or signature recognition problems. Because this particular protection problem can also be considered as a current waveform recognition problem, the use of ANN seems to be a good choice. The ANN-based approach can detect healthy operating based on recognizing their wave shapes, more precisely, by differentiating them from the fault current wave shapes. It gives a trip signal in the case of internal fault only and exercises restraint under healthy, magnetizing inrush and over-excitation conditions.

In power systems, transformer is one of the essential elements and thus transformer protection is of critical importance. There are a variety of protective relays to provide reliable and secure transformer protection. Although using the second harmonic restrain/blocking approach that was used in the past may prevent false tripping during inrush conditions, practice has shown that it may sometimes increase fault clearance time for heavy internal faults followed by CT saturation. Since the CT iron-core has a non-linear characteristic it saturates at high currents, or when DC is present in the primary current.

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 358–365, 2011.
© Springer-Verlag Berlin Heidelberg 2011

Now-a-days, differential relays are widely used to detect the internal faults of a transformer which involves converting the primary current and secondary currents in a common base and comparing them. In principle, this protection scheme makes use of current difference flowing through the different terminals of transformer so as to distinguish between internal fault and healthy operating conditions. The transformer differential relay should be designed such that it does not mal-operate during magnetizing inrush and over excitation conditions which fools differential relay [1]. Some researchers have investigated the use of ANN methods for the same purpose [5, 6, 9, 10].

In this work, an attempt has been made to design a novel relay using ANN which can differentiate faulty operating condition from other healthy operating conditions based on the shape or pattern of the differential current signal. A two layered feed forward back propagation network has been developed to realise the relay. This network consists of 48 inputs, 6 outputs. After experimenting with different number of hidden neurons, a network with 85 hidden neurons is finalised. Detailed description about these inputs and outputs is given in later sections. The ANN-based pattern recognition algorithm has been tested to evaluate the performance of the proposed method in terms of accuracy and speed, and the encouraging results were obtained.

2 Power System Simulation for Pattern Generation

A three-phase 220/110 kV power system including a 150 km transmission line has been used to produce the required test and training patterns. The simulation was done by means of Simulink (MATLAB version R2010b) software. Table 1 represents the associated data with this power system model. The condition system has been produced, as shown in Table 1, using this system to train the ANN. Faults are located at different points of transmission line. Also, they involve inrush current and over excitation condition with different voltage angles and with different loads. Breakers are connected at different positions for creating different operating conditions.

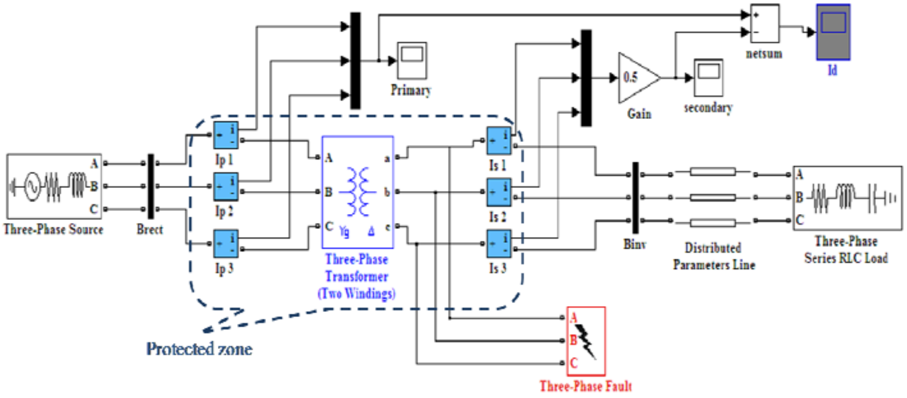


Fig. 1. Simulated three-phase power system model

Table 1. Condition system for Patterns Data Generation

Condition system	*Normal:* When there is no fault. *Internal fault*: 3-phase fault is connected to Transformer secondary. *Inrush*: Breaker is closed at different voltage angles *Over-excitation:* At different over voltages
Load(MW)	200, 400, 600, 800 and 1000

3 Neural Network Design and Simulation

The developed ANN, the used network architecture is multi-layered with one input layer, one hidden layer, and one output layer. All layers are fully connected and of the feed forward type. The outputs are nonlinear functions of the input, and are controlled by weights that are computed during learning process. The used learning process is the supervised type, and the used learning paradigm is the back propagation (BP). The basic idea of this algorithm is using sensitivity (gradient) of the error with respect to the weights, to conveniently modify it during iteration steps. The paradigm realizes an optimum nonlinear mapping by associating the input training patterns to the output training patterns via the successive solutions to linear system of equations. This algorithm is developed on the basis of the mean square error (MSE) method.

3.1 Network Architecture and Training

The first step to formulate the problem is identification of proper input and output set. Various architectures and combination of input sets were attempted to arrive at the final configuration with a goal of maximum accuracy. Keeping the number of outputs fixed at 6, the number of input neurons and the number of hidden neurons are varied, on trial and error basis, until it produced minimum error. Each of the differential currents (of each phase) is typically represented in discrete form as a set of 16 uniformly spaced (in time) samples obtained over a data window of one cycle i.e. at the sampling rate of 16 samples per cycle or 800Hz. These samples are used for training and testing the developed ANN. Only one hidden layer is considered to simplify the architecture and number of neurons is varied from 20 to 120 to arrive at the final architecture.

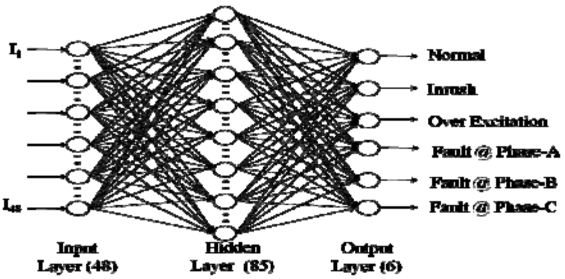


Fig. 2. ANN architecture

Table 2. Training function and parameters

Adaption learning function	learngdm	**Hidden / Output layer transfer function**	Tansig
Training function	Trainlm	**Performance function**	MSE
Epochs	1000	**Mu_dec**	0.1
Goal	0	**Mu_inc**	10
Max-fail	100	**Mu-max**	10^8
Min_grad	1e-10	**Show**	25
Mu	0.001	**Time**	Inf

LEARNGDM- Gradient descent with momentum weight and bias learning function; TRAINLM- Levenberg –Marquardit back-propagation training function.

After enough experimentation, a network with one hidden layer of 85 has been finalized, as shown in Fig. 2. All the architectures were trained for 1000 epochs. The training functions and parameters are given in Table 2.

A software *nprtool* (Neural Network pattern recognition tool, a graphical user interface) in MATLAB SIMULINK has been used for training process. Since the network has to distinguish among four kinds of signals, 4 sets of example signals (cases) have been obtained for this purpose. These cases are normal, magnetizing inrush, over- excitation and internal fault currents.

A total of 1836 training sets of samples (918 sets for healthy conditions and 918 sets for different types of faults, by simulating for a period of 400mSec, i.e. 20 cycles) generated by SIMULINK in MATLAB, have been used to train and test the neural network. Out of these 1836 training sets of samples, 10% samples are used for validation and another 10% are used for testing purpose during training process.

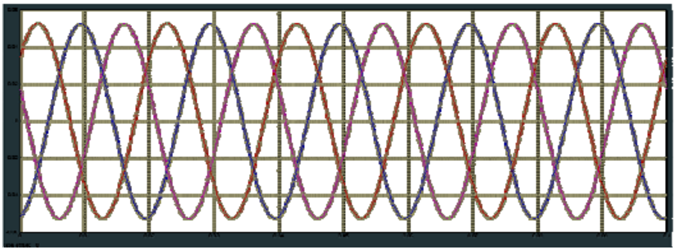


Fig. 3. (a). Normal condition

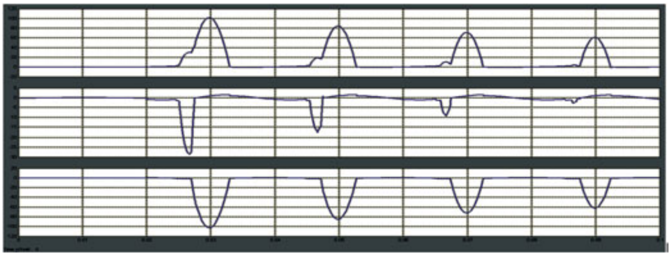


Fig. 3. (b). Inrush condition

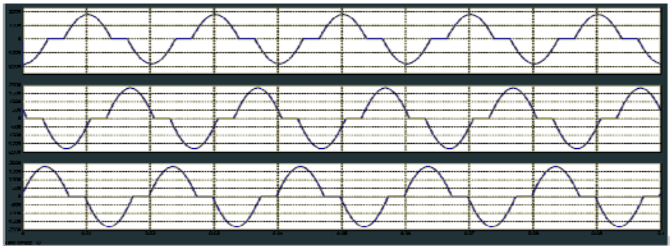


Fig. 3. (c). Over-excitation condition

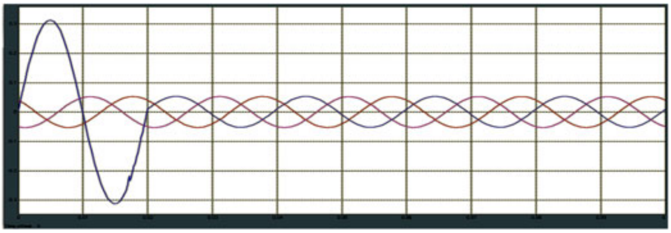


Fig. 3. (d). Internal fault condition (L-G Fault)

The network consists of 48 inputs. These 48 input values are combination of 3 sets of data, each set comprising of 16 differential current samples from each phase. For each case, signals are sampled at the sampling rate of 16 samples per cycle. This data is fed to the ANN in moving window format, i.e. first set of input consists of 1 to 16 samples, 2nd set consists of 2 to 17 samples and so on. Hence, at any instant of time, the ANN will have the latest full cycle data from each phase. The simulated waveform patterns that are used to train the ANN are presented in Fig.3.

On the other side, the ANN will generate 6 outputs, each for 6 different operating conditions, namely, :Normal:, :Inrush:, :Over excitation:, :Fault @ Phase-A:, :Fault @ Phase-B: and :Fault @ Phase-C:. The values of these outputs are always either ‘1’ or ‘0’, ideally. Hence the outputs of the network have a unique set (*e.g.*, 100000 = normal, 010000 = inrush, 001000 = over-excitation, 000100 = L-G Fault at phase-A, 000011 = 2L-G fault at phases B and C). This network monitors all the conditions occurring in the power transformer and it issues the trip signal only in the case of internal fault condition *i.e.* when output is anyone from ‘000100’ to ‘000111’.

3.2 Testing algorithm

Once the training process is completed the network is ready for testing. The network is then fed with new samples that are not used for training. The proposed fault detection algorithm is shown below in the form of flowchart. Before confirming the operating condition as faulty condition and issuing a trip signal, the algorithm first checks for any possible occurrence of inrush and over excitation conditions i.e., for cases that produce an output like ‘010010’ (inrush and fault @ phase B) or ‘001100’ (Over excitation and Fault @ phase A). This verification process is only to improve the

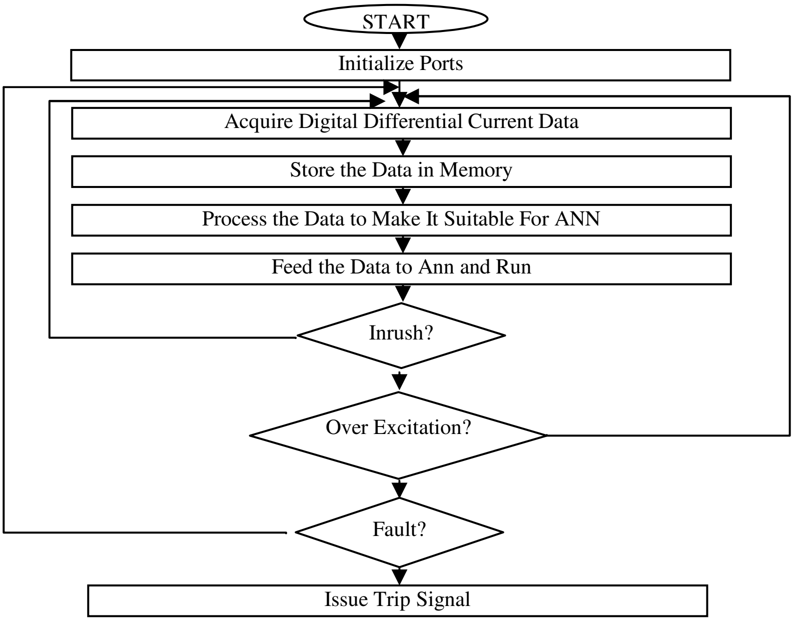

START
Initialize Ports
Acquire Digital Differential Current Data
Store the Data in Memory
Process the Data to Make It Suitable For ANN
Feed the Data to Ann and Run
Inrush?
Over Excitation?
Fault?
Issue Trip Signal

Fig. 4. Flowchart of the proposed algorithm

reliability of the system. In the actual case, the above mentioned cases have very less probability to occur, as the proposed algorithm successfully differentiates the fault conditions from other operating conditions.

4 Network Performance and Numerical Results

Concerning the ANN architecture, parameters such as the number of inputs to the network and the number of neurons in the hidden layer were decided empirically. This process involves experimentation with various network configurations. The learning process was terminated after 1000 epochs. The best validation performance was 0.0019446 for the proposed network and was within acceptable limit. The proposed network responds in a very adequate way, performing the discrimination among normal, inrush, over-excitation, and internal fault currents correctly for all cases. Fig.5 shows the performance graphs and Table 3 shows the performance results of different topologies.

Another important thing in ANN based protection scheme is its speed of issuing trip signal with maximum accuracy. The proposed network produced an output of ‘1’ when the 10th sample of the fault pattern is detected. That means it takes just above half cycle period to detect the fault. However, taking the repeatability and accuracy of the proposed method into consideration, a value more than 0.999 can be considered as acceptable value to issue the trip signal. This is produced when 6th or 7th sample of the fault pattern is detected, which is less than a half cycle period. The tested output

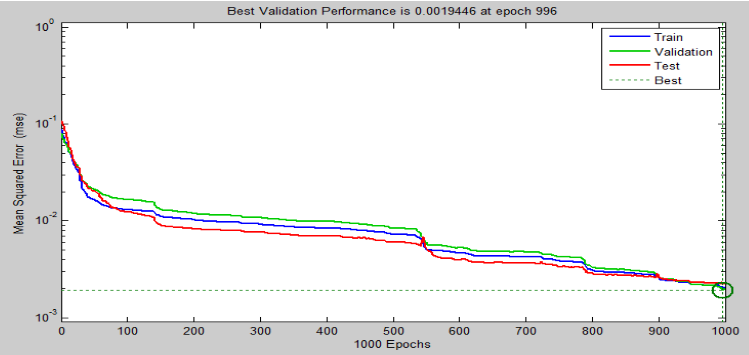


Fig. 5. Learning performance over 1000 epochs

Table 3. Performance of ANN with 48 inputs, 6 outputs, and variable no. of hidden

ANN topology	Best validation performance
48_20_6	0.0847823
48_40_6	0.0152743
48_60_6	0.0090561
48_85_6	0.0019446
48_ 96_6	0.0041062
48_ 120_6	0.0100366

Table 4. Tested output for both the architectures

Operating condition	Outputs							
	1		2		3		4 / 5 / 6	
	O	T	O	T	O	T	O	T
Normal	0.99938	1	0.00017	0	0.00036	0	0.01952	0
Inrush	0.00085	0	1	1	0.00021	0	0.00046	0
Over-excitation	0.00526	0	0.00011	0	0.99994	1	0.00772	0
Internal fault (any phase)	0.00916	0	0.00018	0	0.00392	0	0.99921	1

T = Target ; O = actual output;

results for different operating conditions when 6th sample of fault or inrush or over excitation pattern is detected is shown in Table 4.

5 Conclusion

In this paper a feed forward back propagation neural network model for protection of three-phase power transformer have been proposed. The 48_85_6 architecture could correctly discriminate among the different conditions in power transformer such as normal, magnetizing inrush, over-excitation and internal fault conditions. The FFBPNN based pattern recognition method is efficient in solving classification problems and a differential relay can be considered as a classifier which identifies what kind of event occurs on the network. The ANN has been trained for all the possible sets of simulated data under different operating conditions of transformer. ANN based differential relaying for power transformer shows promising security, accuracy and speed.

References

1. Rahmati, A.: Pattern recognition methods for improvement of differential protection in power transformers. InTech (October 2009)
2. Tripathy, M., Maheshwari, R.P., Verma, H.K.: Neuro-fuzzy Technique for Power Transformer Protection. Electric Power Components and Systems 36, 299–316 (2008)
3. Saha, M.M., Rosolowski, E., Izykowski, J.: Artificial Intelligent Application to Power System Protection
4. Balaga, H., Vishwakarma, D.N., Sinha, A.: Numerical Differential Protection of Power Transformer using ANN as a Pattern Classifier. In: International Conference on Power, Control and Embedded Systems (ICPCES-2010), November 28-December 1, IEEE (2010)
5. Zaman, M.R., Rahman, M.A.: Experimental testing of the artificial neural network based protection of power transformers. IEEE Trans. Power Delivery 13(2), 510–517 (1998)
6. Bastard, P., Meunier, M., Regal, H.: Neural network-based algorithm for power transformer differential relays. IEE Proc. Generat. Transm. Distrib. 142(4), 386–392 (1995)
7. Sadeghierad, M., Teherie Asbagh, A, Monsef, H.: A new algorithm for protection of three-phase power transformers using newral networks. In: AEE 2005 Proceedings of the 4th WSEAS International Conference on Applications of Electrical Engineering (2005)
8. Pihler, J., Dolinar, D.: Improved Operation of Power Transformer Protection Using Artificial Neural Network. IEEE Transctions on Power Delivery 12(3) (July 1997)
9. Kasztenny, B., Rosolowski, E.: Multi-Objective Optimization of a Neural Network based Differential Relay for Power Transformers. In: IEEE Conference on Transmission and Distribution, vol. 2, pp. 476–481 (1999)
10. Zaman, M.R., Rahman, M.A.: Experimental Testing of the Artificial Neural Network Based Protection of Power Transformers. IEEE Transactions on Power Delivery 13(2) (April 1998)
11. Moravej, Z., Vishwakarma, D.N., Singh, S.P.: ANN Based protection Scheme for Power Transformer. Electric Machine and Power Systems 28, 875–884 (2000)
12. Moravej, Z., Vishwakarma, D.N.: ANN Based Harmonic Restraint Differential Protection of Power Transformer. IE(I) Journal-EL 84 (June 2003)
13. Khorashadi-Zadeh, H.: Power Transformer Differential Protection Scheme Based on Symmetrical Component and Artificial Neural Network. In: 7th Seminar on Neural Network Applications in Electrical Engineering, September 23-25. IEEE (2004)
14. Segatto, E.C., Coury, D.V.: A Differential Relay For Power Transformers Using Intelligent Tools. IEEE Transactions on Power System 21(3) (August 2006)
15. Coury, D.V., Segatto, E.C.: Pattern Recognition to Distinguish Magnetizing Inrush from Internal Faults in Power Transformers. In: 4th WSEAS Int. Conf. on Soft Computing, Optimization, Simulation & Manufacturing Systems (SOSM 2004) (2004)

Modified Radial Basis Function Network for Brain Tumor Classification

S.N. Deepa and B. Aruna Devi[1]

[1] Department of EEE, Anna University of Technology, Coimbatore,
Jothipuram P.O, Coimbatore - 641037, India
arunaamurthy@gmail.com

Abstract. The study proposes a modified RBF with better network learning, convergence, error rates and classification results which involves spatial information data points using Gaussian Mixture Model (GMM) and Expectation Maximization (EM) algorithm for automatic biomedical brain tumour detection. The model was used to predict the brain tumour type (benign or malignant). The results showed outperformance of GMM-EM model with spatial points than the standard RBF model.A classification with a success of 85% and 90.3% has been obtained by the classifiers for RBF and RBF-GMM model.

1 Introduction

Human analysis is intuitive and subjective. Computers perform objective and quantitative analysis. The application of machine learning techniques to automatic analysis of biomedical images is in high demand. Magnetic resonance imaging (MRI) is often the medical imaging method of choice when characterization of brain tissues is essential for brain tumor. An alternate to Multilayer Perceptron network and Back Propagation Network is Radial basis functions which have been extensively used for interpolation, regression and classification due to their universal approximation properties and simple parameter estimation.

1.1 Problem Definition

Brain tissues classification into cancerous or non-cancerous automatically is imperative for research and clinical diagnosis. Use of artificial neural network expel in the supervised techniques .One such network is Radial basis neural networks. Data identical distribution is indistinct in regression models. The data in two given sets are self-sufficient when characterizing it. But in case of spatial data, the information is reliant on each other. In traditional RBF, spatial information is ignored. Only identical distribution is measured [7]. Spatial auto-correlation is incorporated in the model of RBF network and investigation of the algorithm in brain tumor classification analysis is to be performed.

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 366–371, 2011.
© Springer-Verlag Berlin Heidelberg 2011

2 Proposed Modified Radial Basis Function Network (MRBFN) for Brain Tumor Classification

The contextual and structural data is not obtained after the segmentation and classification stage[1]. In spectral classification, it is inferred only to which class it belongs to. To yield good recognition of intrinsic properties (i.e. attributes such as size, shape, and length) and semantic knowledge (i.e. spatial relations among different objects of data) spatial information is to be incorporated in the model. Spatial and temporal prior knowledge can be incorporated using Gaussian processes. It can be accurately modelled via a Gaussian Mixture Model (GMM) i.e., data distribution can be expressed as a mixture of multiple normal distributions and spatial points [6]. Expectation Maximization (EM)is used for density estimation of data points in an unsupervised setting, where the membership of any given point to the clusters is not complete and can be fractional and where ability to deal with missing data and observe unidentified variables.

2.1 Proposed MRBFN Algorithm

Given a sufficiently large number M of Gaussian kernels and the freedom to adjust center μ_m and width σ_m for each kernel, RBF networks can achieve arbitrarily small error. A mixture model in input space assumes that data distribution p(x) to be represented as a linear combination of H kernels $\Phi_j(x)$ for conventional RBF as

$$p(x) = \sum_{j=1}^{H} \Phi_j(x) \tag{1}$$

For the proposed RBF, the model is

$$p(x) = \sum_{j=1}^{H} P(j)\Phi_j(x) \tag{2}$$

where $0 \leq P(j) < 1$ are mixing coefficients (weighted)that satisfy the constraint.

$$\sum_{j=1}^{H} P(j) = 1 \tag{3}$$

GMM is a parametric model used to estimate a continuous probability density function from a set of multi-dimensional feature observations [2]. The GMM of *K* multidimensional Gaussian components given by for conventional is:

$$p(x) = \sum_{k=1}^{k} N(x \mid \mu_k, \Sigma_k) \tag{4}$$

For the modified weighted RBF it is given by

$$p(x) = \sum_{k=1}^{k} \pi_k N(x \mid \mu_k, \Sigma_k) \tag{5}$$

where π_k, μ_k and $\sum_k$ are mixing coefficients, mean and covariance respectively. The parameters of GMM are estimated by Expectation Maximization (EM) algorithm[2].

The steps to carry out the EM algorithm, (E step) are as follows:

1) Initialize the means μ_k and co variances $\sum_k$ and mixing coefficients π_k.
2) Evaluate the initial value of the log likelihood.
3) Evaluate the responsibilities using the current parameter values:

$$\gamma(z_{nk}) = \frac{\pi_k (x_n \mid \mu_k, z_k)}{\sum_{j=1}^{K} \pi_j N(x_n \mid \mu_j, \sum_j)} \tag{6}$$

The steps to carry out the EM algorithm, (M step) are as follows:

1) Initialize the means μ_k and co variances $\sum_k$ and mixing coefficients π_k.
2) Evaluate the initial value of the log likelihood.
3) Re-estimate the parameters using the current responsibilities:

$$\mu_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n \tag{7}$$

$$\sum_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k^{new})(x_n - \mu_k^{new})^T \tag{8}$$

$$\pi_k^{new} = \frac{N_k}{N} \text{ where } N_k = \sum_{n=1}^{N} \gamma(z_{nk}) \tag{9}$$

4) Evaluates log likelihood and check for convergence of either the parameters or the log likelihood:

$$\ln p(X \mid \mu, \Sigma, \pi) = \sum_{n=1}^{N} \ln\{\sum_{k=1}^{K} \pi_k N(x_n \mid \mu_k, \Sigma_k)\} \tag{10}$$

5) If the convergence criterion is not satisfied, return to the E step.

The algorithm minimizes the squared difference between outputs and their corresponding target values using a gradient descent procedure. For a given pair{**x**, **t**} of input **x** and its corresponding target t **,** a constrained optimization network error is applied, which penalizes small radii [5]. The optimization objective is:

$$E = \frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{M}(y_k^n - t_n^k)^2 + \alpha\sum_{k=1}^{M}\frac{1}{r_k} \tag{11}$$

where r_k is the radius of cluster k. α is a small regularizing parameter, estimated by cross validation on the training data. Constraint covariance matrices for the GMM, uses the diagonal or spherical covariance or full covariance types [3].The gradient descent algorithm error as in (9) is employed. The constraint is spatial autocorrelation exists, that is, y_i is not only affected by its own x_i, but also by x_j and y_j of its neighbors $s_j \in N(s_i)$. RBF coupled map lattice is used as spatial temporal predictor to model the brain tumour image classification to detect cancerous portion. The input is fused by weighted averaging each site and its neighbors. When P(j) and π_k is 0, in (1) and (7), system is purely temporal i.e. stronger the spatial effect , larger π_k or α will be used.

2.2 MRBFN for Brain Tumor Classification

Feature selection reduces the dimensionality of feature space, removes redundant, irrelevant, or noisy data. The sequence of steps to image classification includes histogram equalization, binarization, morphological operations, and region isolation and feature extraction. A set of statistical invariant features implemented are:

Table 1. Satistical Features Evaluated

Entropy	$\sum_{i=1}^{N}\sum_{j=1}^{N}(\frac{\mu_{(i,j)}}{R})\log(\frac{\mu_{(i,j)}}{R})$
Homogeneity	$\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{[\frac{ijp(i,j)}{R}]}{1+(i-j)^2} \quad \text{for } i \neq j$
Contrast	$\sum_{i=1}^{N}\sum_{j=1}^{N}(i-j)^2(\frac{P(i,j)}{R})$
Energy	$\sum_{i=1}^{N}\sum_{j=1}^{N}(\frac{\mu(i,j)}{R})^2$
Correlation coefficient	$\sum_{i-0}^{G-1}\sum_{j=0}^{G-1}\frac{ijp(i,j)-\mu_x\mu_y}{\sigma_x\sigma_y}$

The specimen images for training and testing the neural was gathered from PSG Hospitals, Coimbatore. The images are digitalized in 512x512; 12bit per pixel from MR Avanto 1.5T MRI scanner. Table 2 denotes the performance accuracy rates. Around 72 real time image datasets were used for training and testing.

Table 2. Performance accuracy rates for the classifiers

Method	TP	TN	FP	FN	Sensitivity %	Specificity %	Accuracy %
RBF	22	29	1	8	0.73	0.967	0.85
MRBF	27	29	1	3	0.9	0.96	0.903

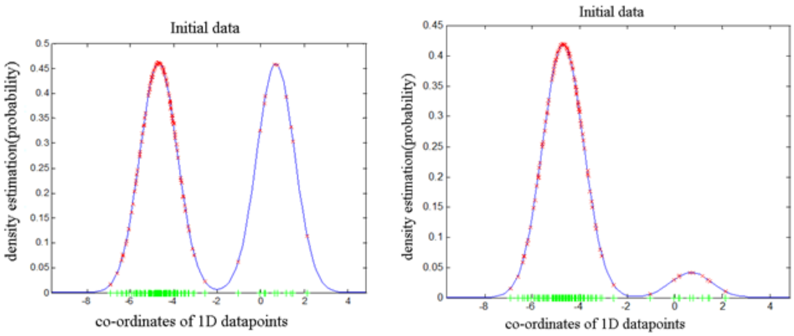


Fig. 1. Density Estimation for RBFN and MRBFN

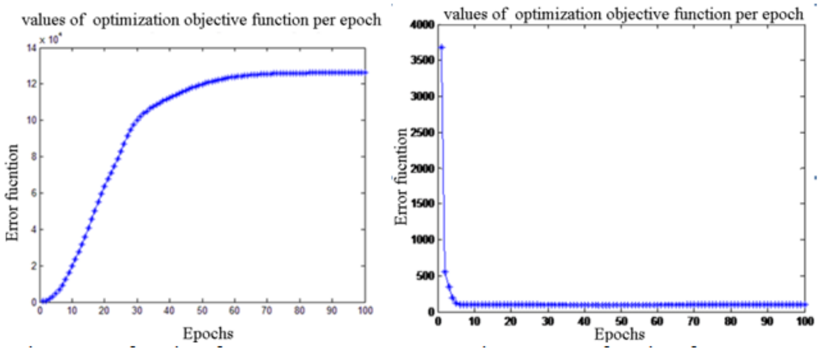


Fig. 2. Error function for RBFN and MRBFN

3 Discussion

Spatial information is ignored in conventional RBF networks which fail to give satisfactory results on spatial data. Spatial information fusion into RBF networks by fusing output from hidden and output layers is done by π_k or α (or weighted or mixing factor) coefficient. MSE keeps decreasing as α grows within [0, 1] and it achieves the lowest value around α =1.Once α exceeds 1, MSE soon increases sharply at a larger rate than its previous decreasing rate. The error of the cost objective cost estimation in MRBFN converges faster as in figure 2. In case of LMSE error, in conventional RBF the error co-efficient increases monotonically and converges after more number of iterations. Inclusion of spatial data points, increase the probability density function as in figure 1 when compared to conventional RBF with only identical distribution points.

4 Conclusion

This paper shows MRBF, with respect to algorithm design, the main distinction of the proposed modified RBFN is inclusion of α, with its weights from 0 to 1. Further geometrical properties like moments, can be implemented to further classify the grades of tumor along with cystic component and solid component with white matter and grey matter. There are other locations where spatial information can be pushed into RBF networks, for instance, the center selection which will prove more superior in performance. The modified RBFN outperforms in full covariance because the principal axes of the basis functions is parallel to that of its feature space in diagonal covariance

Acknowledgments. The authors would like to thank PSG Hospitals for collection of MRI images. The authors extend their gratitude to Dr. R.Shanmugasundaram, PSG Hospitals for his valuable ideas and support in consideration of MRI images.

References

1. Alexandridis, A., Sarimveis, H., Bafas, G.: A new algorithm for online structure and parameter adaptation of RBF networks. Neural Networks 16(17), 1003–1017 (2003)
2. Juba, D., Varshney, A.: Modelling and Rendering Large Volume Data with Gaussian Radial Basis Functions. Technical Report, University of Maryland (2007)
3. Hartman, E.J., Keller, J.D., Kowalski, J.M.: Layered neural networks with gaussian hidden units as universal approximations. Neural Computation 2(2), 210–215 (1990)
4. Orr, M.J.L: Introduction to radial basis function networks. Technical report, Center for Cognitive Science, University of Edinburgh (1996)
5. Padmavathy, J.: A Comparative study on Breast Cancer Prediction using RBF and MLP. Int. Jou. Scientific & Engg. Research 2(1) (2011)
6. Cohen, S., Nathan: Global Optimization of RBF Networks. IEEE Transaction on Neural Networks 7(4), 969–984 (1996)
7. Hu, T., Sung, S.Y.: Data Fusion in Radial Basis Function Networks for Spatial Regression. Neural Processing Letters 21, 81–93 (2005)

Attribute Clustering and Dimensionality Reduction Based on In/Out Degree of Attributes in Dependency Graph

Asit Kumar Das[1], Jaya Sil[2], and Santanu Phadikar[3]

[1,2] Department of Computer Science and Technology,
Bengal Engineering and Science University, Shibpur Howrah, India
[3] Department of Computer Science and Engineering, West Bengal University of Technology,
Saltlake, Kolkata -64, India
`{akdas,js}@cs.becs.ac.in, sphadikar@yahoo.com`

Abstract. In order to mine useful information from huge datasets development of appropriate tools and techniques are needed to organize and evaluate such data. However, ultra high dimensionality of data poses serious challenges in data mining research. The method proposed in the paper encompasses a new strategy in dimensionality reduction by attribute clustering based on the dependency graph of the attributes. Information gain, an established theory of measuring uncertainty and quantified the information contained in the system, of each attribute is calculated that expresses dependency relationship between the attributes in the graph. The underlying principles able to select the optimum set of attributes, called reduct able to classify the dataset as could be done in presence of all attributes. The rate of dimension reduction of the datasets of UCI repository is measured and compared with existing methods and also the classification accuracy with reduced dataset is calculated by various classifiers to measure the effectiveness of the method.

1 Introduction

Data mining [9, 10] is the process of discovering non-trivial, previously unknown and potentially useful information from large volume of data [8, 9]. Data mining research becomes popular as the scale of data has increased due to the advent of data processing and computing technology [1, 3]. Data preprocessing is an essential step to develop efficient data mining techniques, which are able to forecast future trends and activities in support of the mankind.

Feature selection [2, 3] is an important preprocessing step applied to reduce dimensionality of data and complexity of data analysis. Rough set based reducts [4, 5], statistical methods [6, 7] and correlation-based methods [20] have contributed towards improving attribute reduction techniques over quite a long period of time. Statistical methods are generally lower in computational complexity compared to the rough reducts and the correlation-based methods. However, there is no loss of information in generating reducts, resulting better accuracy in prediction.

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 372–380, 2011.
© Springer-Verlag Berlin Heidelberg 2011

Feature selection effectively reduces dimensionality of data by removing redundant attributes resulting improvement in classification accuracy. The goal of feature selection is to avoid selecting too many or too few features than is necessary. If too few features are selected, there is a high chance that the information content in the set of features is low. On the other hand, too many (irrelevant) features consist of noise that may overshadow the relevant information required for classification. Hence, a trade-off is essential that must be addressed while developing algorithms for feature selection so that classification accuracy is improved by reducing the influence of noise.

In the proposed method information gain, an established theory of measuring uncertainty and quantified the information contained in the system, of each attribute is calculated that expresses dependency relationship between the attributes. An information gain (IG) table consisting of IG of attributes is constructed expressing dependency relationship among the attributes. The table is reduced by removing the attributes don't have significant influence to classify the objects. Functional dependencies of these attributes are used to construct a dependency graph [11]. Vertices of the graph represent attributes while edges represent the degree (in-degree / out-degree) of dependency of the attributes. Depending on the in-degree/out-degree values, the attributes are clustered [18, 19] and elements (attributes) of each cluster are sorted in ascending / descending order. Within each cluster the attributes are further grouped considering their ordered in-degree/out-degree value and score of each group is computed, which signifies that each attribute in same group are related to each other. The overall process is repeated for dataset in each group. The process terminates when each group contains either exactly one attribute or no further grouping is possible. Since, the attributes in a group are dependent with each other and independent with the attributes in other group, selecting arbitrarily single attribute from each cluster provide a compact set of attribute called reduct.

In the experiment, the rate of dimension reduction of the datasets collected from UCI machine learning repository [17] is measured and compared with existing methods [14, 15] and also the classification accuracies with reduced datasets obtained by proposed as well as existing methods are calculated by various classifiers [9, 10] to measure the effectiveness of our method.

The paper is organized as follows: Proposed Functional Dependency (FD) among the attributes and their degree of dependencies are described in section 2. Single reduct generation based on partitioning of the attributes is discussed in sedction 3. Experimental results obtained for various benchmark datasets are described in Section 4 and finally, conclusion is summarized in section 5.

2 Functional Dependency of the Attributes

Information gain is the concept, applied intuitively towards reduction of uncertainty in classifying objects with respect to some given attributes. Uncertainty is measured using information entropy that quantifies the expected value of the information contained in the attributes of objects. Uncertainty relative to a given value of an attribute (feature) is called conditional entropy. The entropy $H(X)$ of a variable X is defined in equation (1) and conditional entropy H(X/Y) is referred as entropy of *X* after observation of variable *Y*, given in equation (2).

$$H(X) = \sum_i P(x_i) \log P(x_i) \tag{1}$$

$$H\left(\frac{X}{Y}\right) = \sum_j p(y_j) \sum_i p\left(\frac{x_i}{y_j}\right) log\, p\left(\frac{x_i}{y_j}\right) \tag{2}$$

Where $P(x_i)$ is the prior probability of i^{th} value of *X*, $P(X_i/Y_j)$ is the post priori probability of *X* with the given values of y_j where $j = 1, 2, .., N$; number of variables.

The information gain of a given attribute *X* with respect to attribute *Y* is denoted by IG(X/Y) and described as the difference between the entropy and conditional entropy, defined in equation (3).

$$IG\left(\frac{X}{Y}\right) = H(X) - H\left(\frac{X}{Y}\right) \tag{3}$$

According to the measure of information gain, a feature *Y* is regarded to be more correlated to feature *X* than to feature *Z*, if IG(X/Y)>IG(Z/Y). Symmetry is a desired property for correlations between features. However, information gain is biased in favor of features with higher values and so by using the information gain metric it is easier to remove the redundancy in the datasets.

Information gain (IG) is used to compute functional dependency of each attribute with respect to others, based on which, the dependency graph is constructed. Each attribute of the dataset is mapped as vertex of the graph while edges represent dependency relationship among the attributes. In-degree and out-degree of the vertices are calculated denoting degree of dependency of an attribute on other attribute and takes an important role to partition the attribute set. The algorithm to compute the functional dependency among the attributes is described below:

```
Algorithm: Functional_Dependency_of_Attributes (DS, IG)
 Begin
  u=1/*Compute functional dependencies of attributes */
   For J = 1 to N do
    {sum = 0;
     For I = 1 to N do
       {If ( I != J)
         sum = sum + IG (I,J); /* sum of column    value
                                      of each attribute */
        }  /* end of I loop */
      avg=sum/(N-1);/*average value of each attribute*/
         /*Compute the attribute dependency matrix FD*/
      For I = 1 to N do
       {If ( I!= J ) then
          { If ( IG(I,J) > avg ) then
       /*gain of I^th attrib given J^th attrib > avg gain*/
             {v = 1;
              FD[u][v]=J;/*calculation of FD*/
              v + +;/*increment the column value */
```

```
                FD[u][v] = I ;
                u++;    /* increment the row value */
               }
             }
         }    /* end of I loop */
     }     /* end of J loop */
  End

  Algorithm: Degree_of_Dependency (DS, FD)
  /*Compute in & out degree of attrib in DS from FD and
store in 1st and 2nd columns array deg[][]respectively */
  Begin
    u=u-1; /*u is the number of Functional dependencies*/
    For I = 1 to N do
       {/* compute in and out degree for each attribute*/
        deg[I][1] = deg[I][2] = 0;
       For J = 1 to u - 1 do
         {/*compute the in and out degree of I^th attrib*/
           If( FD[J][1] = = I) then       /*out-degree*/
               deg[I][1]++;
           If( FD[J][2] = = I)     /*in-degree*/
               deg[I][2]++;
          }   /* end of J loop */
         }    /* end of I loop */
  End
```

3 Partitioning of Attributes and Reduct Formation

The attributes with same in-degree are grouped together, partitioning the attributes with their in-degree values; $CLUS_{in\text{-}degree} = \{IN\_GR_1, IN\_GR_2, \ldots, IN\_GR_m\}$, where the groups are arranged in ascending order based on their in-degree. Similarly, the groups are arranged in descending order with respect to their out-degree and obtain the cluster $CLUS_{out\text{-}degree} = \{OUT\_GR_1, OUT\_GR_2, \ldots, OUT\_GR_n\}$. Such ordering of the groups implies that higher the out-degree of a vertex (attribute), more attributes are dependent on that attribute and so it provides important information about the system. Similarly, lesser the in-degree value of a vertex means corresponding attribute has less dependency on other attributes and so provide valuable information about the system.

The rank functions Rf_{in} and Rf_{out} are defined on the domain sets $CLUS_{in\text{-}degree}$ and $CLUS_{out\text{-}degree}$ to map the groups in the clusters using the indices, as given below in equation (4) and (5).

$Rf_{in}(x) = I_x$ (4) where, $x \in CLUS_{in\text{-}degree}$ and I_x is the index of x in $CLUS_{in\text{-}degree}$

$Rf_{out}(x) = I_y$ (5) where, $y \in CLUS_{out\text{-}degree}$ and I_y is the index of y in $CLUS_{out\text{-}degree}$

Based on the rank of the element, score of each attribute is computed using equation (6) where for each attribute A_i in A, it's associated groups g_1 and g_2 with respect to $CLUS_{\text{in-degree}}$ and $CLUS_{\text{out-degree}}$ are obtained.

$$Score(A_i) = \frac{Rf_{in}(g_1) + Rf_{out}(g_2)}{2} \tag{6}$$

Thus, equal importance is given to both the in-degree and out-degree of the attributes to measure their score value. The attributes with same score are correlated with each other and so, based on scores the attributes are partitioned as described by the algorithm below:

```
Algorithm: Partition_based_on_Attribute_Score(SCORE,CLUS)
  Begin /*Partition of attrib according to their score*/
     CLUS = Ø  /*2-D array, contains cluster of attrib*/
     row = 1;
     While(1)
     {/*select minimum score*/
        min = First non-negative value in array Score;
       For I = 1 to N do
         { If ((min > SCORE[I]) && (SCORE[I] > 0)) then
              min = SCORE[I];
         }
           If (min = = -1) then
             break;/*partitioning done, outside of loop*/
        For I=1 to N do /*this loop compute one cluster*/
         { If (SCORE[I] = = min) then
            {
               SCORE[I] = -1;
              CLUS [row] = CLUS [row] ∪ {A_I}
            }
         }
       row = row + 1;
      }    /* end of while loop */
    No_in_clstr = row - 1; /*number of clusters in CLUS*/
  End
```

Each new partition is attempted for partitioning again and the process is repeated until no more partitioning is possible. Thus, the final partitions are either singleton set or set of attributes. Then, the reduct is obtained for the decision system by combining all the singleton partitions and one arbitrarily selected attribute from each of the rest partitions.

4 Results and Discussions

The proposed method computes a single reduct for the bench mark datasets selected from UCI machine learning repository [13]. At first, all the numeric attributes are discretized by ChiMerge [16] discretization algorithm. To measure the efficiency of the method, *k*-fold cross-validations, where *k* ranges from 1 to 10 are carried out on the reduced datasets and classified using “Weka” tool [12]. The method is illustrated with *iris* dataset.

The *iris* dataset consists of 150 objects and 5 attributes. The last column represents the decision attribute with 3 class labels Iris-setosa, Iris-virginia and Iris-veritosa while remaining attributes *Sepal Length*, *Sepal width*, *Petal Length* and *Petal width* represent condition attributes of the sample decision system, renamed for simplicity as *A*, *B*, *C* and *D* respectively. For each condition attribute *X*, individual entropy *H*(*X*) is computed, using equation (1). Next, the conditional entropy for each attribute is calculated using equation (2) and conditional information gain IG(*X*/*Y*) is calculated by equation (3), described in Table 1.

Table 1. Conditional information gain for attributes

Given Attribute Attribute	A	B	C	D
A	×	0.064	0.207	0.203
B	0.045	×	0.098	0.114
C	0.191	0.143	×	0.355
D	0.191	0.033	0.345	×
Average information gain of the system	**0.142**	**0.08**	**0.217**	**0.224**

From the information gain (IG) table (Table 1), the average information gain of the system is calculated. In particular, if attribute *A* is known, then the average information gain of the system is 0.142. However, the conditional information gain of attribute *B* is 0.045, which implies that for known attribute *A*, *B* does not gain much information. Therefore, *B* is not dependent on *A*. But attributes *C* and *D* gain much information because respective IG value is more than the average information gain of the system. So *C* and *D* are dependent on *A*. Thus, $A \rightarrow C$ and $A \rightarrow D$ are obtained. Similarly, for other known attributes, dependencies are computed and the following attribute dependencies are obtained: $A \rightarrow C$, $A \rightarrow D$, $B \rightarrow C$, $C \rightarrow D$, $D \rightarrow C$.

Now the dependency graph G = (V, E) is constructed so that attributes in the dependency diagram are the vertex set *V* and there is an edge $E(v_i, v_j)$ if and only if $v_i \rightarrow v_j$. Fig. 1 shows the dependency graph. Now, from the graph, in-degree and out-degree of each attribute are computed as shown in Table 2 and attributes are clustered based on their in-degrees and out-degrees.

Table 2. Degree of Dependency

vertices	outdegree	indegree
A	2	0
B	1	0
C	1	3
D	1	2

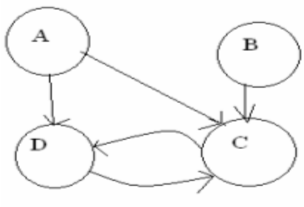


Fig. 1. Attribute Dependency graph

Here, two clusters $CLUS_{in\text{-}degree}$ and $CLUS_{out\text{-}degree}$ are obtained where, $CLUS_{in\text{-}degree}$ = {{AB}, {D}, {C}} and $CLUS_{out\text{-}degree}$ ={{A}, {BCD}}. Then, the score of each attribute in a group is calculated using equation (6), where, score (A) = 1, score (B) = 1.5, score (C) = 3.5 and score (D) = 2.5. Since, score of each attribute is distinct, four partitions each with single attribute is obtained which are: {A}, {B}, {D}, {C}.

The proposed method and well known dimensionality reduction methods such as 'Cfs Subset Eval' (CFS) method [14], 'Consistensy Subset Evaluator' (CON) method [15] are applied on various datasets and results are listed in Table 3. It shows that the rate of dimensionality reduction is better in the proposed method in most of the datasets compare to CFS and CON methods. The method is not only reducing more attributes but also it is observed that the accuracy of various classifiers for reduced dataset by proposed method is compatible with that obtained by CFS and CON methods.

Table 3. Attribute reduction and accuracy comparison for different datasets

Datasets (# attributes)	Methods (reduced # attributes)	Classifiers				
		PART	Bagging	MCS	Naïve Bayes	Average
Iris (4)	Proposed (4)	94	94	94	96	94.50
	Cfs (2)	91	94	93	92	92.50
	Con (2)	91	94	93	92	92.50
Wine (13)	Proposed (6)	96	97	98	98	97.25
	Cfs (11)	93	95	98	97	95.50
	Con (5)	94	95	96	97	95.50
Heart (13)	Proposed (8)	77	83	84	89	83.25
	Cfs (7)	76	81	84	86	81.75
	Con (10)	74	81	84	84	80.75
Credit (15)	Proposed (9)	88	91	89	76	86.00
	Cfs (7)	85	84	87	75	82.75
	Con (13)	86	85	86	76	83.25
Vehicle (18)	Proposed (11)	68	69	71	53	65.25
	Cfs (11)	63	66	65	48	60.50
	Con (7)	65	66	64	42	59.25

5 Conclusion

Data Mining is not the answer to all problems and sometimes it has been over emphasized. It is expensive to carry out the entire process and therefore has to be thought out clearly. Feature selection approaches reduce the complexity of the overall process by allowing the data mining system to focus on what is really important. Thus, the data mining knowledge produced is found more meaningful. Also the new users / end users will get better results quickly. The proposed method encompasses a new strategy in dimensionality reduction by attribute clustering based on the dependency graph of the attributes. The method strongly based on the graph theory and information gain, an established theory of measuring uncertainty and quantified the information contained in the system. The rate of dimension reduction of the datasets of UCI repository is measured and compared with existing methods and also the classification accuracy with reduced dataset is calculated by various classifiers to measure the effectiveness of the method.

References

1. Baldonado Pal, S.K., Mitra, S.: Neuro-Fuzzy pattern Recognition: Methods in Soft Computing. Willey, New York (1999)
2. Carreira-Perpinan, M.A.: A review of dimension reduction techniques. Technical report CS-96-09, Department of Computer Science, University of Sheffield (1997)
3. An, A., Huang, Y., Huang, X., Cercone, N.J.: Feature Selection with Rough Sets for Web Page Classification. In: Peters, J.F., Skowron, A., Dubois, D., Grzymała-Busse, J.W., Inuiguchi, M., Polkowski, L. (eds.) Transactions on Rough Sets II. LNCS, vol. 3135, pp. 1–13. Springer, Heidelberg (2004)
4. Pawlak, Z.: Rough sets. International Journal of information and Computer Sciences 11, 341–356 (1982)
5. Pawlak, Z.: Rough set theory and its applications to data analysis. Cybernetics and Systems 29(1998), 661–688 (1998)
6. Gupta, S.C., Kapoor, V.K.: Fundamental of Mathematical Statistics. Sultan Chand & Sons, A.S. Printing Press, India (1994)
7. Devroye, L., Gyorfi, L., Lugosi, G.: A Probabilistic Theory of Pattern Recognition. Springer, New York (1996)
8. Devijver, P.A., Kittler, J.: Pattern Recognition: A Statistical Approach. Prentice-Hall, Englewood Cliffs (1992)
9. Han, J., Kamber, M.: Data Mining: Concepts and Techniques, MK (2001)
10. Witten, I.H., Frank, E.: Data Mining:Practical Machine Learning Tools and Techniques with Java Implementations, MK (2000)
11. Deo, N.: Graph Theory with Applications to Engineering and Computer Science. Prentice-Hall of India Pvt. (1995) ISBN-81-203-0145-5
12. WEKA: Machine Learning Software, `http://www.cs.waikato.ac.nz/~ml/`
13. Murphy, P., Aha, W.: UCI repository of machine learning databases (1996), `http://www.ics.uci.edu/mlearn/MLRepository.html`
14. Hall, M.A.: Correlation-Based Feature Selection for Machine Learning PhD thesis, Dept. of Computer Science, Univ. of Waikato, Hamilton, New Zealand (1998)
15. Liu, H., Setiono, R.: A Probabilistic Approach to Feature Selection: A Filter Solution. In: Proc.13th Int'l Conf. Machine Learning, pp. 319–327 (1996)
16. Kerber, R.: ChiMerge: Discretization of Numeric Attributes. In: Proceedings of AAAI 1992, Ninth Int'l Conf. Artificial Intelligence, pp. 123–128. AAAI Press (1992)

17. Daren, Y., Qinghua, H., Wen, B.: Combining multiple neural networks for classification based on rough set reduction. In: IEEE int. Conf. Neural Network & Signal Processing, Nanjing, China, December 14-17 (2003)
18. Jain, A., Murty, M., Flynn, P.: Data clustering: A review. ACM Comput. Surv. 31(3), 264–323 (1999)
19. Everitt, B., Landau, S., Leese, M.: Cluster Analysis. Arnold, London (2001)
20. Hall, M.A.: Correlation-Based Feature Selection for Machine Learning PhD thesis, Dept. of Computer Science, Univ. of Waikato, Hamilton, New Zealand (1998)

MCDM Based Project Selection by F-AHP & VIKOR

Tuli Bakshi[1], Arindam Sinharay[2], Bijan Sarkar[3], and Subir kumar Sanyal[4]

[1] Jadavpur University, Kolkata, India
`tuli.bakshi@gmail.com`
[2] Department of Information Technology,
Future Institute of Engineering & Management, Kolkata, India
`arindam.sinharay@gmail.com`
[3] Department of Production Engineering, Jadavpur University, Kolkata, India
`bijon_sarkar@email.com`
[4] Department of Production Engineering, Jadavpur University, Kolkata, India
`sanyal_s_k@yahoo.co.in`

Abstract. The multiple criteria decision-making method (MCDM) VIKOR is based as the aggregating function representing "closeness to the ideal", which originate in the compromise programming method. Linear normalization is used in VIKOR to eliminate the units of criterion functions. The VIKOR method preceded by Fuzzy AHP which usually calculates the weights of criteria in continuity, determines a compromise solution, presenting the maximum "group utility" for the "majority" and a minimum of an individual regret for the "opponent". In this paper the authors have shown single objective optimization model first, then balance it with multiple objectives in management process. As for case study, construction project management has been considered. It involves objectives as duration, cost, quality, resource leveling etc. On the basis of that, the paper has proposed an integrated fuzzy multi-objective optimization model of duration, cost, quality and resource leveling.

Keywords: MCDM, VIKOR, Compromise Solution, Fuzzy AHP.

1 Introduction

Many papers have proposed analytical models as aids in conflict resolving situations. Among various approaches available for conflict management, one of most prevalent is multi-criteria decision making (MCDM). It may be considered as a complex and dynamic process having engineering level [1] and managerial level. The engineering level of MCDM process defines alternative criteria and points out the consequences of selecting any one of them from the viewpoint of various criteria. The level also performs multi-criteria ranking alternatives. Managerial level defines the final optimal alternative at which public official called "decision makers" have the power to accept or reject the solution proposed by engineering level [2-3].

MCDM is the approach dealing with ranking and selection of one or more vendors from a pool of providers in the current case study. Therefore, an effective framework

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 381–388, 2011.
© Springer-Verlag Berlin Heidelberg 2011

for comparison based on the evaluation of multiple conflicting criteria is the need. MCDM has been one of the fastest growing areas of project selection and management domain, as it is often directed that many concrete problems can be represented by several conflicting criteria. The decision process of selecting appropriate alternatives usually deal with many factors for instance organizational needs, goals, risks, benefits and limited resources etc. A compromise solution for a problem with conflicting criteria can yield the decision maker to reach the final decision. The foundation for compromise solution was established by [2] and [4] . The VIKOR method was proposed as one applicable technique to implement within MCDM [5].

In this paper, Analytic hierarchy process (AHP) which is improved under fuzziness to find the weight [6] of the criteria. Then VIKOR method have been used on modeling aggregating function by linear normalization[7], in order to reveal the best compromise solution.

2 Model Establishment

Duration, cost and quality, resource leveling are some criteria, which considered as objectives and involves in construction project management. It is obvious to integrate and make a equilibrium multi-objective in management process, while in engineering practice decision maker tend to analyze an optimize time and cost objective. Researchers now-a-days have made interesting contribution on time, cost, quality and resource leveling equilibrium. Some of them, emphasized on single objective quality and resource leveling model. As it is known if single objective deals with shortest duration, lowest cost, highest quality and smallest resource leveling coefficients, so the ultimate goal is to construct a multi-criteria optimization model. On this basis the current researchers proposed a MCDM model as follows :

$$\phi_1 = \text{min (Duration)}, \quad \phi_2 = \text{min (cost)}, \quad \phi_3 = \text{min (time)}, \qquad \phi_4 = \text{max (Quality)}$$

Further characterizing this multi-objective model the current researchers define four sub objective models which conflict with each other. So it is necessary to choose a scheme which make the sub objective close to the optimum. This paper deals with Fuzzy AHP[8] and VIKOR[9]-[10] method to solve the above problem.

3 Fuzzy-AHP Method

The fuzzy AHP technique can be viewed as an advanced analytical method developed from the traditional AHP. According to the method of Chang's (1992) [8] extent analysis, each criterion is taken and extent analysis for each criterion, g_i 's performed on, respectively. Therefore, m extent analysis values for each criterion can be obtained by using following notation :

$M_{g_i}^{1}$, $M_{g_i}^{2}$, $M_{g_i}^{3}$, $M_{g_i}^{4}$, $M_{g_i}^{5}$ .... $M_{g_i}^{m}$, Where g_i is the goal set (i = 1,2,3,4,..................n) and all $M_{g_i}^{j}$ (j = 1,2,3,4,........m) are Triangular Fuzzy Numbers(Tens). The steps of the analysis can be given as follows:

Step 1:- The fuzzy synthetic extent value (S_i) with respect to the i^th^ criterion is defined as equation (1):-

$$S_i = \sum_{j=1}^{m} M_{g_i}^{j} \otimes (1 / [\sum_{i=1}^{n} \sum_{j=1}^{m} M_{g_i}^{j}]) \tag{1}$$

To obtain (2) :-

$$\sum_{j=1}^{m} M_{g_i}^{j} \tag{2}$$

Perform the fuzzy addition operation of m extent analysis values for a particular matrix given in equation (3) below, at the end step of calculation, new (l, m, and u) set is obtained and used for the next

$$\sum_{j=1}^{m} M_{g_i}^{j} = (\sum_{j=1}^{m} l_j, \sum_{j=1}^{m} m_j, \sum_{j=1}^{m} u_j) \tag{3}$$

Where l is the lower limit value, m is the most promising value and u is the upper limit value and to obtain (4)

$$(1 / [\sum_{i=1}^{n} \sum_{j=1}^{m} M_{g_i}^{j}]) \tag{4}$$

Perform the fuzzy addition operation of $M_{g_i}^{j}$ (j =1, 2, 3, 4 ...m) values given as equation (5)

$$\sum_{i=1}^{n} \sum_{j=1}^{m} M_{g_i}^{j} = (\sum_{i=1}^{n} l_i, \sum_{i=1}^{n} m_i, \sum_{i=1}^{n} u_i) \tag{5}$$

and then compute the inverse of the vector in the equation (5) and equation (6) is then obtained as:-

$$(1 / [\sum_{i=1}^{n} \sum_{j=1}^{m} M_{g_i}^{j}]) = [\frac{1}{\sum_{i=1}^{n} u_i}, \frac{1}{\sum_{i=1}^{n} m_i}, \frac{1}{\sum_{i=1}^{n} l_i}] \tag{6}$$

Step 2:- The degree of possibility of $M_2 = (l_2, m_2, u_2) \geq M_1 = (l_1, m_1, u_1)$ is defined as equation (7),

$$V(M_2 \geq M_1) = \sup_{y \geq x} [\min (\mu_{M1}(x), \mu_{M2}(y))] \tag{7}$$

And x and y are the values on the axis of membership function of each criterion. This equation can be written as : $V(M_2 \geq M_1) = 1$, if $m_2 \geq m_1$

$$= 0, \quad \text{if } l_1 \geq u_2 \qquad = \frac{l_1 - u_2}{(m_2 - u_2) - (m_1 - l_1)}, \text{otherwise} \tag{8}$$

Step 3:-The degree possibility for a convex fuzzy number to be greater than k convex fuzzy numbers M_i (i= 1,2,3,4,5...................k) can be defined by

V (M $\geq M_1, M_2, M_3 \ldots\ldots\ldots\ldots\ldots\ldots M_k$) = min V (M$\geq M_i$),i = 1, 2 ...k. Assume that equation (9) is $d^*(A_i)$ = min V ($S_i \geq S_k$) (9)

For k = 1, 2, 3.................n; k ≠ i. Then the weight vector is given by equation (10):-

W^* =($d^*(A_1)$, $d^*(A_2)$,............. $d^*(A_n)$)^T (10)

Where A_i (i = 1, 2, 3 ...n) are n elements.

Step 4:-Via normalization, the normalized weight vectors are given in equation 11:-

W = (d (A_1), d (A_2), d (A_3) ...d (A_n))^T (11)

Where W is non-fuzzy numbers.

4 VIKOR Method

The VIKOR introduces the multi-criteria ranking index based on the particular measure of "closeness" to the "ideal" solution.The compromise ranking algorithm VIKOR has the following steps:

Step1: Determine the best f_i^* and the worst f_i^- values of all criterion functions, i = 1,2,.......n. If the ith function represents a benefit then:

$$f_i^* = \max_j \; f_{ij} \quad , \quad f_i^- = \min_j \; f_{ij}$$

Step 2: Calculate the values of S_j and R_j where j = i,2, ...j (No. of alternatives)

$$S_j = \sum_{i=1}^{n} w_i (f_i^* - f_{ij}) / (f_i^* - f_i^-) \, , \; R_j = \max[w_i (f_i^* - f_{ij}) / (f_i^* - f_i^-)]$$

where w_i = weight of ith criteria, expressing their relative importance.

Step 3: Compute the values Q_j , j = 1,2,.........j, by the relation

$$Q_j = v(S_j - S^*) / (S^- - S^*) + (1 - v)(R_j - R^*) / (R^- - R^*)$$

where $S^* = \min_j \; S_j \quad S^- = \max_j \; S_j$,

$$R^* = \min_J \; R_j \quad R^- = \max_j R_j$$

v= the weight of strategy of maximum group utility,where as (1-v) is the individual regret, here v = 0.5.

Step 4: Rank the alternatives, sorting by the values of S_j, R_j and Q_j, in descending order. The results are three ranking lists.

Step 5: Propose as a compromise solution.

5 Case Study

According to expert's decision, the following matrix is formed and then by using Triangular Fuzzy Number the Fuzzy evaluation matrix is formed [11]

Table 1. Evaluation Matrix

Criteria	D	TC	RLC	Q
D	1	2	1	1
TC	0.5	1	0.5	0.75
RLC	1	2	1	2
Q	1	1.33	0.5	1

Table 2. Fuzzy Evaluation Matrix

Criteria	D	TC	RLC	Q
D	(1,1,1)	(1,2,3)	(0.75,1,1.25)	(0.75,1,1.25)
TC	(o.33,0.5,1)	(1,1,1)	(0.25,0.5,0.75)	(0.5,0.75,1)
RLC	(0.8,1,1.33)	(1.33,2,4)	(1,1,1)	(1,2,3)
Q	(0.8,1,1.33)	(1,1.33,2)	(0.33,0.5,1)	(1,1,1)

Now calculating all the values by applying Chang's theory the following results are obtained: $S_D = (3.5, 5, 6.5) \otimes (0.04, 0.057, 0.078) = (0.14, 0.28, 0.51)$

$S_{Tc} = (2.08, 2.75, 3.75) \otimes (0.04, 0.057, 0.078) = (0.08, 0.16, 0.29)$

$S_{RLC} = (4.13, 6, 9.33) \otimes (0.04, 0.057, 0.078) = (0.17, 0.34, 0.73)$

$S_Q = (3.13, 3.83, 5.33) \otimes (0.04, 0.057, 0.078) = (0.13, 0.22, 0.42)$

$V(S_D \geq S_{Tc}) = 1$, $V(S_D \geq S_{RLC}) = 0.86$, $V(S_D \geq S_Q) = 1$

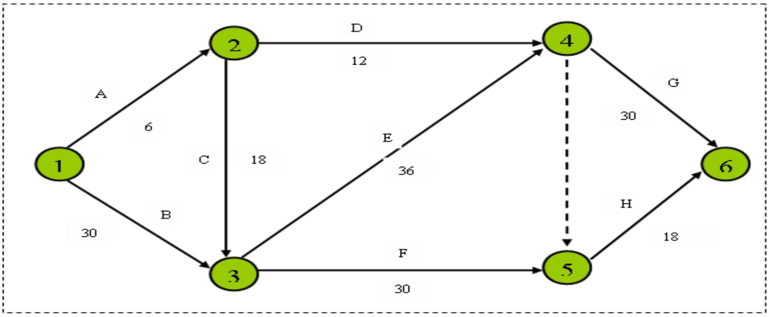
$V(S_{TC} \geq S_D) = 0.54$, $V(S_{TC} \geq S_{RLC}) = 0.41$, $V(S_{TC} \geq S_Q) = 0.73$

$V(S_{RLC} \geq S_D) = 1$, $V(S_{RLC} \geq S_{TC}) = 1$, $V(S_{RLC} \geq S_Q) = 1$

$V(S_Q \geq S_D) = 0.81$, $V(S_Q \geq S_{TC}) = 1$, $V(S_Q \geq S_{RLC}) = 0.67$

Minimum of all values (0.86, 0.41, 1, and 0.67) .The weight W = (0.29, 0.14, 0.34, 0.23).

Schedule planning of a certain project [12] is shown as the Fig.1. The character above the arrow indicates the activity's name and the number under the arrow indicates the duration of each activity. The data of each activity's duration, cost, quality and resource demand are given in Table 1. The relationship between indirect cost and duration is shown in Table2. In this case 7 schemes are chosen of which the duration is 96 days, 84 days, 78 days, 76 days, 70 days,62 days and 58 days to analyze as follows:



Network diagram of project

Fig. 1.

Table 3. Project duration cost, quality, resource & demand matrix

Activity(i,j)	Normal duration			Limited duration			Total demand of rsource
	d^U (day)	DC^U (¥)	q	d^L (day)	DC^L (¥)	q	
A(1,2)	6	1500	1	4	2000	0.85	30
B(1,3)	30	9000	1	20	10000	0.90	450
C(2,3)	18	5000	1	10	6000	0.80	450
D(2,4)	12	4000	1	8	4500	0.70	72
E(3,4)	36	12000	1	22	14000	0.90	180
F(3,5)	30	8500	1	18	9200	0.83	210
G(4,6)	30	9500	1	16	10300	0.95	450
H(5,6)	18	4500	1	10	5000	0.90	256

Table 4. Indirect Cost & Relation Matrix

Total duration (day)	55	60	65	70	75	80	85	90	95	100
Indirect cost(¥)	14640	15240	15840	16440	17040	18240	18440	18840	19240	20040

Table 5. Calculation Result Matrix

Scheme	Duration (day)	Total cost ((¥)	Resource leveling coefficient(RCL)	Quality (Q)
1	96	70391.89	11.46	1
2	84	69655.17	14.06	0.9920
3	78	68871.35	15.02	0.9816
4	76	68870.56	16.29	0.9782
5	70	69044.10	14.94	0.9688
6	62	69739.35	12.38	0.9376
7	58	70001.72	14.85	0.9220

Now we can construct the decision matrix X on basis of VIKOR method as follows:

Table 6. Initial Matrix for VIKOR Method

Weight of criteria	**0.29 (MIN)**	**0.14 (MIN)**	**0.34 (MIN)**	**0.23 (MAX)**
Criteria	**Duration (D)**	**Total Cost (TC)**	**Resource-leveling coefficient(RLC)**	**Quality**
C_1	96	70391.89	11.46	1
C_2	84	69655.17	14.06	0.9920
C_3	78	68871.35	15.02	0.9816
C_4	76	68870.56	16.29	0.9782
C_5	70	69044.10	14.94	0.9688
C_6	62	69739.35	12.38	0.9376
C_7	58	70001.72	14.85	0.9220

Table 7. Normalized Matrix

Weight of criteria	**0.29 (MIN)**	**0.14 (MIN)**	**0.34 (MIN)**	**0.23 (MAX)**
Criteria	**Duration (D)**	**Total Cost (TC)**	**Resource-leveling coefficient(RLC)**	**Quality**
C_1	1	1	0	0
C_2	0.684	0.516	0.538	0.103
C_3	0.526	0.0005	0.737	0.236
C_4	0.474	0	1	0.279
C_5	0.316	0.114	0.720	0.4
C_6	0.105	0.571	0.190	0.8
C_7	0	0.744	0.702	1

Table 8. Weighted Normalized Matrix & Final Result

Criteria	**D**	**TC**	**RLC**	**Q**	S_j	R_j	$\frac{(S_j - S^*)}{(S^- - S^*)}$	$\frac{(R_j - R^*)}{(R^- - R^*)}$	Q_j	**Rank**
C_1	0.29	0.14	0	0	0.43	0.29	0.332	0.679	0.506	3
C_2	0.198	0.072	0.183	0.024	0.477	0.198	0.551	0.090	0.321	6
C_3	0.153	0.00007	0.251	0.054	0.458	0.251	0.463	0.429	0.446	4
C_4	0.137	0	0.34	0.064	0.541	0.34	0.850	1	0.925	1
C_5	0.092	0.016	0.245	0.092	0.445	0.245	0.402	0.391	0.397	5
C_6	0.030	0.080	0.065	0.184	0.359	0.184	0	0	0	7
C_7	0	0.104	0.239	0.23	0.573	0.239	1	0.353	0.677	2

From the final result, the preference order of seven schemes is **C_4 > C_7> C_1> C_3> C_5> C_2> C_6.** Thus optimal scheme is no.4,its duration is 76 days, total cost 68870.56, resource leveling coefficient is 16.29 and quality is 0.9782.So when the organizer construct and allocate resources according to construction schedule of No.4 , the project will gain the maximum profit.

6 Conclusion

A large systematic engineering is needed for making schedule for large scale project under consideration of characteristics such as long duration, large cost, complicated process, easily affected natural condition etc. In balance resource alocation increased engineering cost, delay of duration characterize negative effects on project quality. This paper establishes a multi-objective optimization model using VIKOR method. This method is based on an aggregrating function representing "closeness to the ideal". The VIKOR method introduces the ranking index based on the particular measure of closeness to the ideal solution. This model use linear normalization. The normalized value does not depend on the evaluation unit of a criteria function. Finally, this paper does not consider the trade off involve by normalization in obtaining the aggregating function and this topic remains for further ongoing research.

References

[1] Duckstein, L., Opricovic, S.: Multiobjective optimization in river basin development. Water Resource Research 16(1), 14–20 (1980)

[2] Yu, P.L.: A class of solutions for group decision problems. Management Science 19(8), 936–946 (1973)

[3] Opricovic, S., Tzeng, G.: Compromise solution by MCDM methods: A comparative analysis of VIKOR and TOPSIS. European Journal of Operational Research 156, 445–455 (2004)

[4] Zeleny, M.: Multiple Criteria Decision Making. McGraw-Hill, New York (1982)

[5] Opricovic, S.: Multicriteria Optimization of Civil Engineering Systems. Faculty of Civil Engineering, Belgrade (1998)

[6] Bakshi, T., Sarkar, B.: MCA Based Performance Evaluation of Project Selection. International Journal of Software Engineering & Applications (IJSEA) 2(2), 14–22 (2011)

[7] Bakshi, T., Sinharay, A., Sarkar, B.: Exploratory Analysis of Project Selection through MCDM. In: ICOQM-10, pp. 128–131 (2011)

[8] Chang, D.-Y.: Theory and Methodology Applications of the extent analysis method on fuzzy AHP. European Journal of Operational Research 95, 649–655 (1996)

[9] Pavlicic, D.: Normalization affects the results of MADM methods. Yugoslav Journal of Operations Research (YU-JOR) 11(2), 251–265 (2001)

[10] Olson, D.L.: Comparison of three multicriteria methods to predict know outcomes. European Journal of Operational Research 130(3), 576–587 (2001)

[11] Bakshi, T., Sinharay, A.: An Integrated Novel Approach in MCDM under Fuzziness. International Journal of Computer & Communication Technology 2, 36–43 (2011)

[12] Huang, J., Wang, X., Zhou,Y.: Multi-objective Decision Optimization Schedule Based on Improved TOPSIS, 978-4244-4639-1/09/$25.00@ 2009 IEEE (2009)

Nonlinear Time Series Modeling and Prediction Using Local Variable Weights RBF Network

Garba Inoussa[1] and Usman Babawuro[2]

[1] College of Information Science and Engineering, Central South University, 410083,
Changsha, Hunan, P.R. China
[2] Department of Computer Science, Central South University, 410083,
Changsha, Hunan, P.R. China
garbainoussa@hotmail.com

Abstract. This paper proposes a Local Variable-Weights RBF Network (LVW-RBFN) with the aim to address the problem of modeling and prediction of nonlinear time series. The proposed model is a four layered RBFN comprising of input layer, hidden layer, weight layer and output layer. The LVW-RBFN is an enhanced type of RBF network, in which the constant weights that connect the hidden layer with the output layer in the standard RBFN are replaced by functions of the RBFN's inputs computed via the weight layer. This model has the merit of making usage of more linear parameters and learning the dynamic of nonlinear time series through the hidden and weight layers. An offline optimization technique known as Structured Nonlinear Parameter Optimization Method (SNPOM) was used to estimate the model. Simulation results for the modeling and prediction of some nonlinear time-series show the feasibility and effectiveness of the proposed model.

Keywords: Modeling, Prediction, Nonlinear time series, RBF, Local variable weights.

1 Introduction

Over the past several decades, various models such as Regression Analysis [1], Artificial Neural Network (ANN) [2], Wavelet Neural Network (WNN) [3],Fuzzy Logic (FL) [4], Box-Jenkins (ARMA/ARIMA) [5], and Support Vector Regression (SVR) [1, 6-8], have been developed and explored to address the problems encountered in the field of pattern recognition, system identification and control, nonlinear time series modeling and prediction, function approximation and signal processing, with the ANN being the one gaining most popularity. This has led to the development of many types of ANNs, among which, Radial Basis Function Networks (RBFNs) have emerged as the most widely used because of their fastest training speed, simple topology structure and learning scheme. To enhance the performance of the standard RBFNs, two main approaches have so far been developed. The first approach consists of making usage of identification technique to boost the overall performance of the networks. In this line we can mention the SNPOM by Peng Hui [9] and the Evolving

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 389–398, 2011.
© Springer-Verlag Berlin Heidelberg 2011

Radial Basis function networks by H. Du [10]. The second approach consists of enhancing the RBF networks structure. In this line, Reza Langari et al [11] make usage of the *regression weights* to replace the *constant weights* in the output layer of the RBF networks. This method has the merit of rethinking the weight form of RBFNs. These weights are expressed as a linear combination of input variables plus a free constant and a zero-mean Gaussian white noise. However it requires an extra complex optimization technique to compute the weights. Additionally, one can mention the PG-RBF [12] and normalized PG-RBF [13]. These models have the ability of creating new hidden unit, and detect as well as remove inactive unit. It possesses two scaling parameters which eliminate the symmetry restriction and provide the neurons in the hidden layer with greater flexibility with respect to function approximation.

In this paper, the Local Variable Weights RBF Network (LVW-RBFN) is proposed. The LVW-RBFN is an enhanced type of RBF in which the weights are expressed as a function of the RBFN's inputs. It differs from the regressive weights RBF [11] by the fact that its weights do not contain any error. The error is in fact added to the network output, and each term in each of the weight is taken as a power function with the power being a positive integer number. While, the power of each term in the regressive weights RBF is equal to one. Furthermore, unlike the PG-RBF [12] and normalized PG-RBF [13] whose weights are expressed as a local linear model, and can create new hidden unit, and detect as well as remove inactive unit, the proposed model has its weights expressed as a local nonlinear model and keeps the same number of hidden unit. The structured nonlinear parameter optimization method (SNPOM) Peng Hui [9] is applied to estimate the parameters of the proposed models. The performance and effectiveness of our models are evaluated using Mackey-Glass time series, EEC and the Box-Jenkins Gas Furnace Time series.

The remaining parts of this paper are organized as follows. Section 2, defines the nonlinear time series modeling problem. Section 3, presents LVW-RBFN models. Section 4, introduces the identification technique. The performance of the proposed model and that of some existing models are compared in Section 5 and finally the paper is concluded in section 6.

2 Problem Statement

Considering one-step-ahead prediction problem of a given nonlinear time series $\{x(t) \in \Re, t = 1, 2, ..., N\}$, the fundamental problem is how to construct a function, $f : \Re^d \to \Re^1$ with satisfactory prediction accuracy with the following form

$$x(t) = f(x(t-1), ..., x(t-d)) + \zeta(t) \tag{1}$$

where $f(\bullet)$ is the nonlinear map and $\zeta(t)$ denotes the noise regarded as Gaussian white noise which is independent of observation. Eq 1, can be further written as

$$x(t) = f(\mathbf{X}(t-1)) + \zeta(t) \tag{2}$$

where $\mathbf{X}(t-1)=\left[x(t-1),...,x(t-d)\right]^T$ is a vector of lagged values of $x(t)$ and is regarded as the state vector at time t.

3 Local Variable Weights RBFN

The Local Variable weights RBFN is an enhanced type of RBFN. This new model is a four layered artificial feedforward neural network comprising of input, output, hidden and weight layers. Each node in the hidden layer implements a radial basis function (RBF) and has a parameter vector called center which is used as an argument of the RBF to compare with the network input to produce a radially symmetrical responses. These responses are scaled by local variable weights computed as a function of network inputs. They, all are combined to produce the network output. The LVW-RBFN model makes usage of more linear parameters than a standard RBFN. It has a merit of learning the dynamic of nonlinear time series through the hidden and weight layers. The input output relationship of the LVW-RBFN can be described by the following equations.

$$\begin{cases} x(t)=c_0+\sum_{k=1}^{m}\phi_k(\mathbf{X}(t-1))\Phi(\mathbf{X}(t-1),\mathbf{Z}_k)+\zeta(t) \\ \phi_k(\mathbf{X}(t-1))=\beta_{k,0}+\sum_{i=1}^{d}\beta_{k,i}x(t-i)^{P_{k,i}} \\ \mathbf{Z}_k=(z_{k,1},z_{k,2},...,z_{k,d}) \end{cases} \tag{3}$$

where c_0 represent the bias; m is the number of hidden nodes; d is the dimension of state vector $\mathbf{X}(t-1)$; $\mathbf{Z}_k(k=1,2,...,m)$ are the centers; $p_{k,i}(k=1,...,m;i=1,...d)$ are positive integers; $\phi_k(\mathbf{X}(t-1))$ represents the local variable weights at node k; $\Phi(\mathbf{X}(t-1),\mathbf{Z}_k)$ is the radial basis function at node k and $\beta_{k,i}(k=1,...m;i=0,1,...d)$ are local variable weights coefficients. Different type of functions could be used to approximate the above mentioned radial basis function, but the most common one is the Gaussian function given by

$$\Phi(\mathbf{X}(t-1),\mathbf{Z}_k)=\exp\left\{-\lambda_k\left\|\mathbf{X}(t-1)-\mathbf{Z}_k\right\|_2^2\right\} \tag{4}$$

Where $\lambda_k(k=1,...,m)$ are the scaling parameters, and $\left\|\bullet\right\|_2$ Denotes the vector 2-norm. Introducing the above formula into model (3) yields,

$$x(t)=c_0+\sum_{k=1}^{m}\left(\left(\beta_{k,0}+\sum_{i=1}^{d}\beta_{k,i}x(t-i)^{P_{k,i}}\right)\exp\left\{-\lambda_k\left\|\mathbf{X}(t-1)-\mathbf{Z}_k\right\|_2^2\right\}\right)+\zeta(t) \tag{5}$$

4 Model Identification

The proposed model is identified using the SNPOM as follows.

4.1 Parameter Classification

The set of linear parameters in LVW-RBFN model (5) is:

$$\theta_L = \left\{ c_0, \beta_{k,i} \,\middle|\, k = 1,...m; i = 0,1,...d \right\} \in \Re^{1+m\times(d+1)} \tag{6}$$

and the set of nonlinear parameters is:

$$\theta_N = (\lambda_1, \lambda_2, ..., \lambda_m; Z_1^T, Z_2^T, ..., Z_m^T)^T \in \Re^{m+m\times d} \tag{7}$$

Model (5) can be rewritten as,

$$x(t) = f(\theta_L, \theta_N, \mathbf{X}(t-1)) + \zeta(t) \tag{8}$$

or

$$x(t) = \Psi(\theta_N, \mathbf{X}(t-1))^T \theta_L + \zeta(t) \tag{9}$$

4.2 Initialization

Select the order *m, d* and $p_{k,i}(k = 1,...,m; i = 1,...d)$ by comparing Akaike Information Criterion (AIC) [14] value for different orders and also by looking at the model dynamics and repeat the method for a model with different order before finally selecting the best model.

After the orders of the model are chosen, the initial values $\mathbf{Z}_k^0 (k = 1,2,...,m)$ are chosen randomly from the vector space $\mathbf{X}(t-1)$ then the scaling factor of the model is computed using the following formula.

$$\lambda_k^0 = -\log \varepsilon \Big/ \max\left\{ \left\| \mathbf{X}(t-1) - \mathbf{Z}_k^0 \right\|_2^2 \right\}, \quad \varepsilon \in]0, 0.1]. \tag{10}$$

Note that in this paper ε is taken from $]0, 0.1]$ which is different from $]0.0001, 0.1]$ proposed in the original SNPOM. This is done because for some nonlinear time series such as Box-Jenkins Gas Furnace**,** the SNPOM may require ε to be very close to zero. Model (10) keeps the linear parameter bound and stable when the state $\mathbf{X}(t-1)$ is far from the center.

Fixing $\boldsymbol{\theta}_N^0$ and estimating $\boldsymbol{\theta}_L^0$ by LSM, yields

$$\begin{cases} \hat{\boldsymbol{\theta}}_L^0 = \left[\sum_{t=\tau+1}^{M} \overline{\boldsymbol{\Psi}}_{t,0} \overline{\boldsymbol{\Psi}}_{t,0}^T \right]^{-1} \sum_{t=\tau+1}^{M} \overline{\boldsymbol{\Psi}}_{t,0} \overline{x}(t) \\ \overline{\boldsymbol{\Psi}}_{t,0} = \boldsymbol{\Psi}\left(\hat{\boldsymbol{\theta}}_N^0, \overline{X}(t-1) \right) \end{cases} \tag{11}$$

where $\left\{\overline{x}(i), \overline{X}(i-1) \middle| i=\tau+1, \tau+2, \cdots, M\right\}$ is the measured data-set; τ is the largest time lag of any estimable variable in model (8) or (9); M is the number of measured data observations.

4.3 Optimization

The objective function is taken to be

$$\begin{cases} V(\boldsymbol{\theta}_N, \boldsymbol{\theta}_L) \triangleq \frac{1}{2}\left\|\mathbf{F}(\boldsymbol{\theta}_N, \boldsymbol{\theta}_L)\right\|_2^2 \\ \mathbf{F}(\boldsymbol{\theta}_N, \boldsymbol{\theta}_L) \overset{\Delta}{=} \begin{bmatrix} \hat{x}(\tau+1|\tau) - \overline{x}(\tau+1) \\ \hat{x}(\tau+2|\tau) - \overline{x}(\tau+2) \\ \vdots \\ \hat{x}(M|M) - \overline{x}(M) \end{bmatrix} \end{cases} \tag{12}$$

the optimization problem is to compute

$$\left(\hat{\boldsymbol{\theta}}_N, \hat{\boldsymbol{\theta}}_L\right) = \arg\min_{\theta_N, \theta_L} V(\boldsymbol{\theta}_N, \boldsymbol{\theta}_L) \tag{13}$$

4.4 Parameters Updating

The nonlinear parameter vector $\boldsymbol{\theta}_\mathrm{N}$ is updated by the following formula.

$$\boldsymbol{\theta}_N^{k+1} = \boldsymbol{\theta}_N^k + \delta_k d_k \tag{14}$$

where $k(k=1,2,...,k_{\max})$ denotes the iteration step; d_k is the search direction and δ_k is a scalar step length parameter that gives the distance to the minimum, the search process is based on the LMM, d_k is obtained from a solution of the set of linear equations

$$\left[\mathbf{J}(\boldsymbol{\theta}_N^k)^T \mathbf{J}(\boldsymbol{\theta}_N^k) + \gamma_k \mathbf{I}\right] d_k = -\mathbf{J}(\boldsymbol{\theta}_N^k)^T \mathbf{F}(\boldsymbol{\theta}_N^k, \boldsymbol{\theta}_L^k) \tag{15}$$

where I is the identity matrix and γ_k is a nonnegative damping factor which controls both the magnitude and the direction of d_k. The linear parameter vector $\boldsymbol{\theta}_\mathrm{L}$ is updated by a formula similar to (11) as follows:

$$\begin{cases} \hat{\boldsymbol{\theta}}_L^{k+1} = \left[\sum_{t=\tau+1}^{M} \overline{\boldsymbol{\Psi}}_{t,k+1} \overline{\boldsymbol{\Psi}}_{t,k+1}^T\right]^{-1} \sum_{t=\tau+1}^{M} \overline{\boldsymbol{\Psi}}_{t,k+1} \overline{x}(t) \\ \overline{\boldsymbol{\Psi}}_{t,k+1} = \boldsymbol{\Psi}\left(\hat{\boldsymbol{\theta}}_N^{k+1}, \overline{X}(t-1)\right) \end{cases} \tag{16}$$

The search procedure for determining the step length δ_k in (14) ensures that

$$V(\boldsymbol{\theta}_N^{k+1}, \boldsymbol{\theta}_L^{k+1}) < V(\boldsymbol{\theta}_N^{k}, \boldsymbol{\theta}_L^{k}) \tag{17}$$

at each iteration, where the parameter $\boldsymbol{\theta}_N^{k+1}$ and $\boldsymbol{\theta}_L^{k+1}$ are updated by (14) and (16), respectively. By the way, it may be beneficial to use formula (10) to recomputed the scaling factor λ_k at any time after updating the RBF centers in $\boldsymbol{\theta}_N^{k}$ during the search process, in order to avoid divergence of the linear parameter $\boldsymbol{\theta}_L^{k}$. Hence $\boldsymbol{\theta}_N^{k+1}$ and $\boldsymbol{\theta}_L^{k+1}$ are the best parameters chosen to decrease the objective function (17) at $(k+1)^{th}$ iteration.

5 Results and Discussion

5.1 Modeling and Prediction of Mackey-Glass Time Series

The Mackey-Glass (MG) time series is generated from the following time-delay ordinary differential equation:

$$\frac{dy(t)}{dt} = \frac{ay(t-\tau)}{1+y^c(t-\tau)} - by(t) \tag{18}$$

where $a = 0.2, b = 0.1, c = 10$ and $\tau = 17$ was selected . To stay in conformity with previous works, two thousand data were generated with the initial value $x(0) = 1.2$ based on the fourth-order Runge–Kutta method with time step = 0.1. From the generated data set, 1000 data, from the 118th to the 1117th were extracted. The series obtained is shown in Fig 1. The first 500 data were used for training the model while the remaining 500 were used for testing it. LVW-RBF (10,4) model were used to predict the six-step ahead output of this complex nonlinear time series as follows.

$$\begin{aligned} y(t+6) = c_0 + \sum_{k=1}^{10}\left(\left(\beta_{k,0} + \sum_{i=1}^{4}\beta_{k,i}x(t-6(i-1))^3\right)\exp\left\{-\lambda_k \left\|\mathbf{X}(t-1) - \mathbf{Z}_k\right\|_2^2\right\}\right) \\ + \zeta(t) \end{aligned} \tag{19}$$

$$\mathbf{X}(t-1) = [x(t), x(t-6), x(t-12), x(t-18)]^T$$

The modeling and prediction results are shown in table 1 and the six-step ahead predictive error and the original and fit outputs are shown in Fig 1. It was observed that the proposed model required less hidden neurons thus less centers to produce much better prediction accuracy in comparison with almost all the other methods listed in the above table.

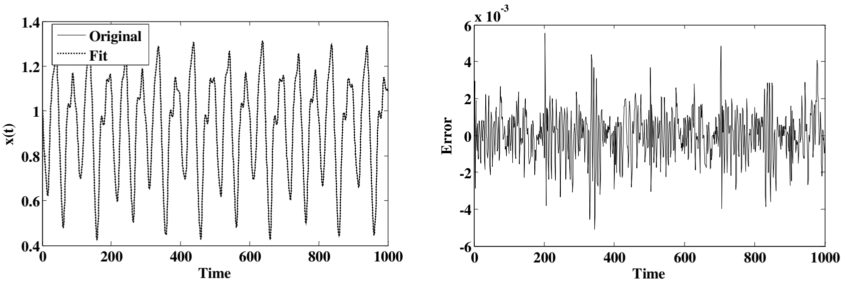


Fig. 1. Six-step ahead prediction (dotted line) and original (solid line) Mackey–Glass series and the predictive error

Table 1. Comparison results for Mackey–Glass time series six-step ahead prediction

Models	Number of neurons	Training set	Testing set
LVW-RBF	**10**	**1.3×10^{-3}**	**1.2×10^{-3}**
LLWNN[3]	10	3.3×10^{-3}	3.6×10^{-3}
RBF-AFS [15]	23	9.6×10^{-3}	1.14×10^{-2}
PG-RBF [13]	12	Not provided	2.87×10^{-3}
FNT [16]	Not provided	6.9×10^{-3}	7.10×10^{-3}
RBF [17]	238	Not provided	1.3×10^{-3}
RBF[10]	98	1.4×10^{-3}	1.5×10^{-3}
MLP[18]	43	Not provided	1.6×10^{-3}
T-S Neuro-fuzzy[18]	16	Not provided	1.5×10^{-3}
RBF[18]	42	Not provided	1.02×10^{-3}
ARIMA–ANN[19]	Not provided	Not provided	2.7×10^{-3}
GFPE [20]	-	Not provided	2.64×10^{-2}

5.2 Modeling and Prediction of EEC Time Series

Electroencephalography known as EEG refers to the recording of the brain's spontaneous electrical activity over a short period of time. To assess the modeling and prediction performance of the proposed models on the EEG, the LVW-RBF (8,5) was used. A set of 630 data was considered, 350 were used for model development and the remaining 280 were used exclusively for testing the model. The MSE obtained between the actual and estimated output using testing set obtained are shown in table 2 and some other results found in the bibliography for this time series are also listed for comparison. The one-step ahead predictive error, the original and fit outputs are shown in Fig 2. It was observed that the LVW-RBF (8, 5) has the smallest Predictive error variance and AIC compared to other models listed in table 2.

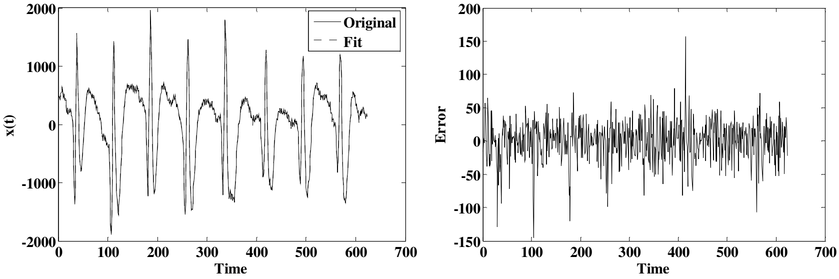


Fig. 2. One-step ahead prediction (dotted line) and original (solid line) EEC and the predictive error

Table 2. Comparison results for EEC time series

Model	Number of centers	MSE	AIC
RBF (5, 8) [21]	40	1.72×10^4	4630
RBF (5, 8) [22]	40	1.32×10^3	6240
Linear AR [21]	-	3.08×10^3	5087
LVW-RBF(5,8)	**40**	$\mathbf{9.47\times10^2}$	**4497**

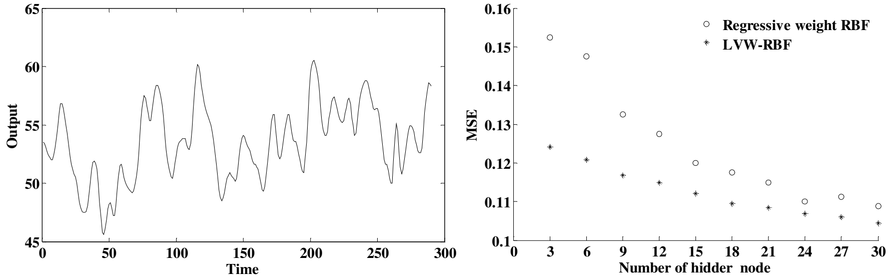


Fig. 3. Box-Jenkins Gas Furnace Time series and MSE per hidden node for LVW-RBF and RW-RBF

5.3 Modeling and Prediction of the Box-Jenkins Gas Furnace Time Series

The Box-Jenkins data was recorded during the combustion process of a methane air mixture. This data set is well known and frequently used as a benchmark for testing nonlinear modeling and prediction methods. The data set consists of 296 pairs of input-output samples with a sample time of 9s. The input $u(t)$ is the gas flow rate into the furnace and the output $x(t)$ is the CO_2 concentrations in the outlet gas, see Fig 3. The performance of our proposed model was assessed by comparing the results obtained by our model and those obtained by the regressive weights RBF proposed by

Reza Langari et al [11]. As in [11], $u(t-4)$ and $x(t-1)$ were taken as input variables and $x(t)$ were taken as output variable and different number of nodes $n = (3, 6, ...30)$ were considered. The mean squared errors, (MSE) are shown in Fig 3 together with the original output. It was observed that, given the same number or some time less number of hidden units, the proposed model achieved better performance than the regressive weights RBF model.

6 Conclusion

In this paper a modeling and predictive approach known as Local Variable Weights RBF network was introduced. The LVW-RBFN is a modified type of RBF network, in which the weights that connect the hidden layer with the output are expressed as special type of functions of RBF's inputs. The LVW-RBFN can be considered as a more general type of RBF. When all the weight regression parameters with exception of the free parameter are equal to zero, the LVW-RBFN is the same as the standard RBF. When all the powers are equal to one, the LVW-RBFN is the same as the regressive weight RBF (RgRBF). The LVW-RBFN learns the nonlinearity from the hidden and weight layers by decomposing the complex, nonlinear system into a set of locally active sub models which are smoothly integrated by their associated Radial basis function. This model has the advantages of having more linear parameters in comparison to the nonlinear parameters; this ameliorates considerably the convergence rate and makes the SNPOM more suitable for identification.

The performance of the proposed method has been assessed through some well-know nonlinear time series and chaotic systems such as Mackey-Glass time series, EEC and the Box-Jenkins Gas Furnace Time series. The results obtained have shown that the LVW-RBF may requires fewer centers to perform better in terms of modeling and prediction accuracy as well as AIC compared to almost all the models listed in this paper. One of the main advantages of the LVW-RBF model over other models listed is that in most of the cases, it requires fewer numbers of centers much more linear parameters to achieve better prediction accuracy and AIC.

References

1. Lin, K., Lin, Q., Zhou, C., Yao, J.: Time Series Prediction Based on Linear Regression and SVR. In: Proc. of the Third International Conference on Natural Computation, pp. 688–691 (November 2007)
2. Boznar, M., Lesjak, M., Mlakar, P.: A Neural Network-Based Method for Short-Term Predictions of Ambient SO2 Concentrations in Highly Polluted Industrial Areas of Complex Terrain. Atmospheric Environment 27(2), 221–230 (1993)
3. Chen, Y., Yang, B., Dong, J.: Time series prediction using a local linear wavelet neural network. Neurocomputing 69, 449–465 (2006)
4. Chen, S.M., Chung, N.Y.: Forecasting Enrollments using High-Order Fuzzy Time Series and Genetic Algorithms. International Journal of Intelligent Systems 21(5), 485–501 (2006)

5. Montañés, E., Quevedo, J.R., Prieto, M.M., Menéndez, C.O.: Forecasting Time Series Combining Machine Learning and Box-Jenkins Time Series. In: Garijo, F.J., Riquelme, J.-C., Toro, M. (eds.) IBERAMIA 2002. LNCS (LNAI), vol. 2527, pp. 491–499. Springer, Heidelberg (2002)
6. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1995)
7. Smola, A.J., Schölkopf, B.: A tutorial on support vector regression. Statistics and Computing 14(3), 199–222 (2004)
8. Drucker, H., Burges, C.J.C., Kaufman, L., Smola, A.J., Vapnik, V.: Support vector regression machines. In: Mozer, M.C., Jordan, M.I., Petsche, T. (eds.) Advances in Neural Information Processing Systems, vol. 9, pp. 155–161. MIT Press, Cambridge (1997)
9. Peng, H., Ozaki, T., Haggan-Ozaki, V., Toyoda, Y.: Structured parameter optimization method for the radial basis function-based state-dependent autoregressive model. International Journal of Systems Science 33, 1087–1098 (2002)
10. Du, H., Zhang, N.: Time series prediction using evolving radial basis function networks with new encoding scheme. Neurocomputing 71, 1388–1400 (2008)
11. Langari, R., Wang, L., Yen, J.: Radial Basis Function Networks, Regression Weights, and the Expectation–Maximization Algorithm. IEEE Trans. on Systems, Man, and Cybernetics—Part A: Systems and Humans 27(5), 613–623 (1997)
12. Rojas, I., Gonzalez, J., Cañas, A., Diaz, A.F., Rojas, F.J., Rodriguez, M.: Short-Term Prediction of Chaotic Time Series by Using RBF Network with Regression Weights. International Journal of Neural Systems 10(5), 353–364 (2000)
13. Rojas, I., Pomares, H., Bernier, J.L., Ortega, J., et al.: Time series analysis using normalized PG-RBF network with regression weights. Neurocomputing 42, 267–285 (2002)
14. Akaike, H.: A new look at the statistical model identification. IEEE Transactions on Automatic Control 19, 716–723 (1974)
15. Cho, K.B., Wang, B.H.: Radial basis function based adaptive fuzzy systems and their applications to system identification and prediction. Fuzzy Sets and Systems 83, 325–329 (1996)
16. Chen, Y., Yang, B., Dong, J., Abraham, A.: Time series forecasting using flexible neural tree model. Information Sciences 174(3-4), 219–235 (2005)
17. Harpham, C., Dawson, C.W.: The effect of different basis functions on a radial basis function network for time series prediction: a comparative study. Neurocomputing 69, 2161–2170 (2006)
18. Gholipour, A., Araabi, B.N., Lucas, C.: Predicting chaotic time series using neural and neurofuzzy models: a comparative study. Neural Processing Letter 24, 217–239 (2006)
19. Valenzuela, O., Rojas, I., Rojas, F., Pomares, H., et al.: Hybridization of intelligent techniques and ARIMA models for time series prediction. Fuzzy Sets and Systems 159, 821–845 (2008)
20. Kim, D., Kim, C.: Forecasting time series with genetic fuzzy predictor ensembles. IEEE Transactions on Fuzzy Systems 5, 523–535 (1997)
21. Shi, Z., Tamura, Y., Ozaki, T.: Nonlinear time series modeling with the radial basis function-based state-dependent autoregressive model. International Journal of System Science 30(7), 717–727 (1999)
22. Ozaki, T., Jimenez, J.C., Peng, H., Ozaki, V.H.: The innovation approach to the identification of nonlinear causal models in time series analysis. In: Brillinger, D.R., Robinson, E.A., Schoenberg, F.P. (eds.) Time Series Analysis and Applications to Geophysical Systems, pp. 195–226. Springer, New York (2004)

Detection of Disease Using Block-Based Unsupervised Natural Plant Leaf Color Image Segmentation

Shitala Prasad[1], Piyush Kumar[2], and Anuj Jain[3]

[1] Department of Electronics and Computer Engineering, Indian Institute of Technology,
Roorkee, Roorkee -247667, India
[2] Department of Information Technology, Indian Institute of Information Technology, Dewghat,
Jhalwa, Allahabad-211012, India
[3] Department of Computer Science and Engineerning,
NRI Institute of Technology and Managment, Baraghata, Gwalior-474001, India
`{sheetala.god.prasad,piyushkumariiita,jaianuj}@gmail.com`

Abstract. A novel unsupervised color image segmentation method is proposed in this paper. First, the image is converted to HSI color model and then it is divided into 5x5 grid matrix resulting into 25 blocks of original leaf image. Each block is then processed separately and passed under an unsupervised segmentation. This segmentation is based on minimizing the energy of each region in the image. This gives the better result in cased of diseased leaf image dataset. This automated system is very much applicable in research work by the botanists specially working with crop diseases and production.

Keywords: Computer Vision, Unsupervised Natural Image Segmentation, Gaussian Mixture Model, Leaf Disease Detection, Histogram-based Segmentation.

1 Introduction

The segmentation issues in colored images are very common but are very useful in many applications of image processing and computer vision. With these segmented images, it is possible to identify different objects or regions in an image using some pattern recognition methods. Colored image segmentations are very important for any type of identification based decision in computer vision and its application is very vast such as patter recognition, image analysis, biometrics, medical imaging re-engineering and training and industries. Image processing is also now very common in agriculture and agricultural products from few decades for the identification and for increasing the production rate in agriculture. Identification of foreground and background in an image is very difficult task and many approaches have been given in [1], [2] [3], [4] and [5].

Unsupervised image segmentation is a simple method of dividing an image into many different regions in such a way that all the regions are homogenous in nature. Some of them are histogram thresholding based techniques [4], color histogram method [5], needs the color information first and then segments the image, but this is not

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 399–406, 2011.
© Springer-Verlag Berlin Heidelberg 2011

useful for the real life applications, such as in our case. And other algorithms are Region-based techniques [6] and the edge-based techniques [7]. In [8], similarity close measure is used for segmentation which works fine for known image but fails for the unknown image. In [9], author used genetic algorithm for real image segmentation caused by the various conditions but is very time taking. Similarly, in [10], graph-based method is proposed that follows the NP-hard idea.

In this image segmentation process Gaussian Mixture Model [11] is used the most mainly for background subtraction in a colored image based on a very simple maximum likelihood (ML) estimation [12]. The segmentation is basically a process of clustering the pixels in an image in some number of groups according to MLE. But in GMM, this number is to be given and thus it is not totally unsupervised segmentation and for which the paper [13] used k-means algorithm to initialize the parameters of GMM algorithm.

An image C, is a composition of two sub-images: one is $C_{forground}$ and other is $C_{background}$ in such a manner that it image C can be composed and decomposed at any time into and from $C_{forground}$ and $C_{background}$ respectively with a blending mask, say γ. Therefore, image C is defined as

$$C = \gamma C_{foreground} + (1-\gamma) C_{background}$$

Further, $C_{forground}$ and $C_{background}$ are finite sets as there can be more than one foregrounds or backgrounds in an image.

$$C_{foreground} = \{C_{foreground1}, C_{foreground2}, \dots C_{foregroundn}\}$$
$$\text{and } C_{background} = \{C_{background1}, C_{background2}, \dots C_{backgroundn}\}$$

In unsupervised image segmentation there are many methods used for the segment of a color image [14] and [15]. They only use the low-level features in an image such as the intensity and the texture features are used for the generation of homogeneity in an image. This leads to bottom-up segmentation strategy causing over-segmentation in the image. Where as in other case, that is in top-down segmentation approach the objects in the image are not well segmented, under-segmentation. The figure 1, shows the detailed information.

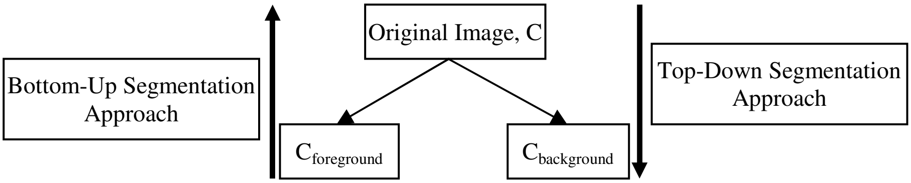

Original Image, C
Bottom-Up Segmentation Approach
Top-Down Segmentation Approach
$C_{foreground}$
$C_{background}$

Fig. 1. Block diagram of segmentation approaches

In this paper, natural plant leaf image is used to detect and recognize the leaf disease the plant has and what possible controls can be used. As it's a natural image and so have no fixed pattern or color-shape, it may vary and thus computer vision system finds difficulty to identify and recognized them. Therefore, an unsupervised segmentation can be used to segment such diseased leafs and then it can be used for the recognition purpose. Here in this paper, a new simple and natural block-based unsupervised segmentation approach is proposed.

The automated leaf disease detection is very helpful for the researches working this area such as botanist and in plant pathology to find out new and old diseases in the plant species. It can also be used in forest research and in agriculture helping farmers to identify the diseases in their plants and control them at the starting stage itself without any expert presence. This system act as and represents the human vision system (HVS) which has an ability to accommodate the color circumstances.

This paper focuses on visual disease segmentation in a digital leaf image. The rest of the paper is organized as: the second section discus about the complete algorithm used in this paper, and in third and fourth section the detailed discussion about the different steps used in the experiment. And in the last fifth section some results and conclusion is given.

2 The Algorithm Overview

The unsupervised natural image segmentation is a challenging problem because of the variance seen in them in regard with color, shape and texture. In an image edge and color are the important discriminate features to analyze an image. The proposed unsupervised segmentation algorithm is simple and performed efficiently with six steps, as shown in figure 2. It is a bottom-up color image segmentation approach applied on the color diseased leaf image.

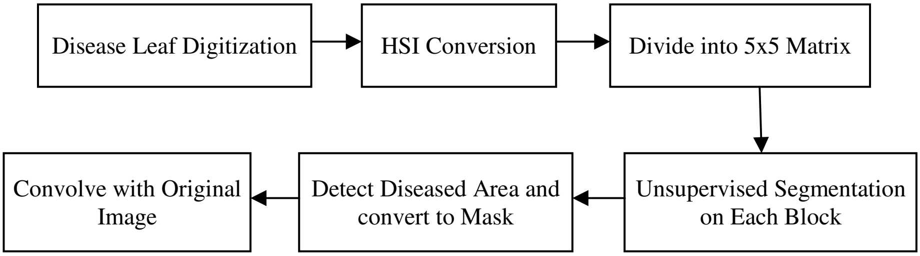

Disease Leaf Digitization
HSI Conversion
Divide into 5x5 Matrix
Unsupervised Segmentation on Each Block
Detect Diseased Area and convert to Mask
Convolve with Original Image

Fig. 2. Block diagram of proposed algorithm

The input to the system is an image of diseased leaf of any species in plant and using a simple unsupervised segmentation method the image is segmented and the output of this system is a diseased area in the leaf having color and texture details both. The detailed procedure is explained in the next section.

3 Unsupervised Segmentation

The first step in this automated leaf disease segmentation is digitization. The plant leaf is digitized using a camera kept with at a fixed height from the ground and the leaf sample is kept on a black background so that they are no need of a background separation task. After this digitalization the image is converted to HSI color mode with the help of Matlab inbuilt function rgb2hsi ('image_name') that uses the color theory formula.

The next step is the division of image into 25 blocks each of equal size as in figure 3. This is because each of the blocks of the image, C have local information and thus will have less noisy data. Because of which the segmentation is easy and accurate. The computation cost in this case will increase a little but it is accepted.

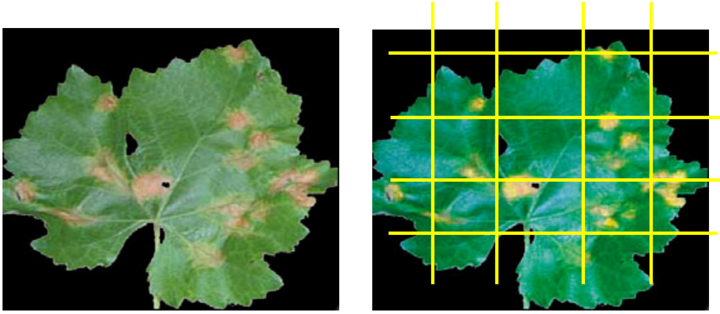


Fig. 3. Right: HSI color image divide into 5x5 block matrix. Left: Original RGB image.

Now each of these sub-images is passed to the unsupervised segmentation stage for the segmentation. The unsupervised segmentation is followed a very simple approach that is minimizing the energy. For example there is two disjoint regions, say $\mathfrak{R}_1$ and $\mathfrak{R}_2$ of an original image $\mathfrak{R}$, such that $\mathfrak{R} = \mathfrak{R}_1 + \mathfrak{R}_2$, where $\mathfrak{R}_i$ and $\mathfrak{R}$ itself are in image domain. Modeling the pixels in both these regions using a statistical method so that it minimized the energy. This can be formulated as below:

$$E = -\int_{\mathfrak{R}_1} log\left(pdf_1(\mathfrak{R}(\ell))\right) d\ell - \int_{\mathfrak{R}_2} log\left(pdf_2(\mathfrak{R}(\ell))\right) d\ell + v * lenght(\mathrm{F})$$

where, pdf_1 and pdf_2 are the Probability Density Function (PDF) of the regions $\mathcal{R}_1$ and $\mathfrak{R}_2$ respectively and the variable F is the frontier between these two regions, that is, region at the edge of a settled area [16]. And E is the energy of that image, $\mathfrak{R}$ in this case. This outputs a segmented image which is unsupervised because in this the PDFs are not fixed and are considered to be unknown and thus re-estimated at each time with the help of the current position of the contour. After the proper segmentation of each block it is recombined and treated as a single image, $\mathrm{C}_{\mathrm{segmented}}$ with segmentations, shown in figure 4 below.

The next important step is to find the diseased region in the segmented image and based on that color value convert it to binary image. Where the diseased region color value is treated as white, valued 1, and rest are valued zero. This binary image shown in figure 5, is used as the mask to be convolved with the original image to fine out the texture feature, discussed in the next section.

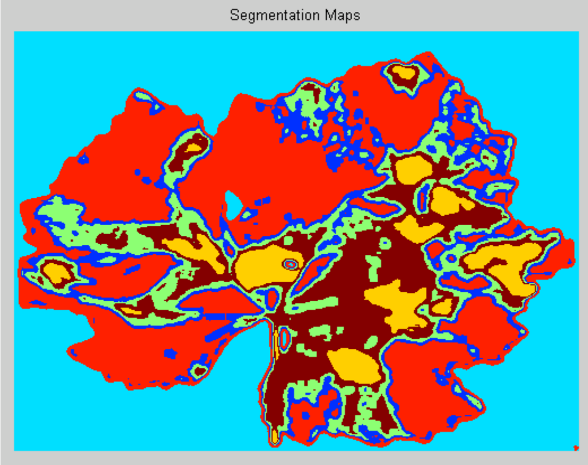


Fig. 4. The segmented image $C_{\text{segmented}}$

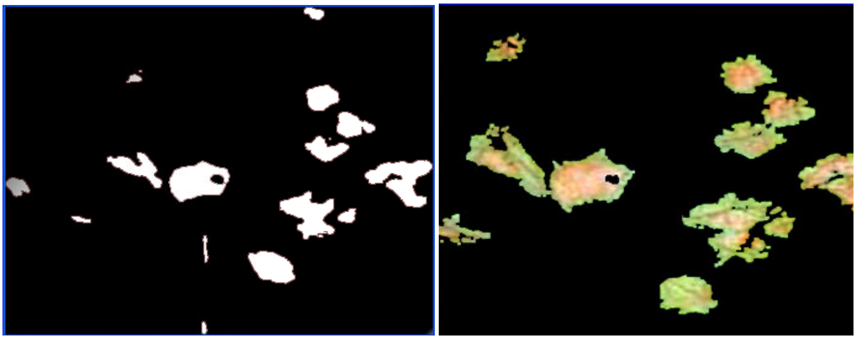


Fig. 5. Left: The mask of the segmented image shown in figure 4. Right: The texture feature of the diseased leaf.

4 Experimental Results

This paper has totally focused on detection of plant leaf diseases using image processing and computer vision. Many experiments are performed on different leaves of different plant species having mainly fungal diseases in leaf. The leaf samples were collected in the lab itself and were processed. To compare the proposed methods performance, some other segmentation techniques were also implemented in the lab whiles the experiments were going on. The other methods implemented were *histogram-based segmentation* and *edge-based segmentation*. The diseases under go for experiments are Tikka disease in groundnut is caused by a fungus Genus Cerocospora Personata or Carachidicola, Powdery Mildew and Downy Mildew are also fungal diseases in many species such as graph, apple, and mango plants which is caused by Erysiphe Necator (causes in graph plant) and Pseudoperonospora Humul, and similarly others diseases are Late Blight and Early Blight in potato and tomato plants and Rust disease in corn and apple species are also fungal diseases.

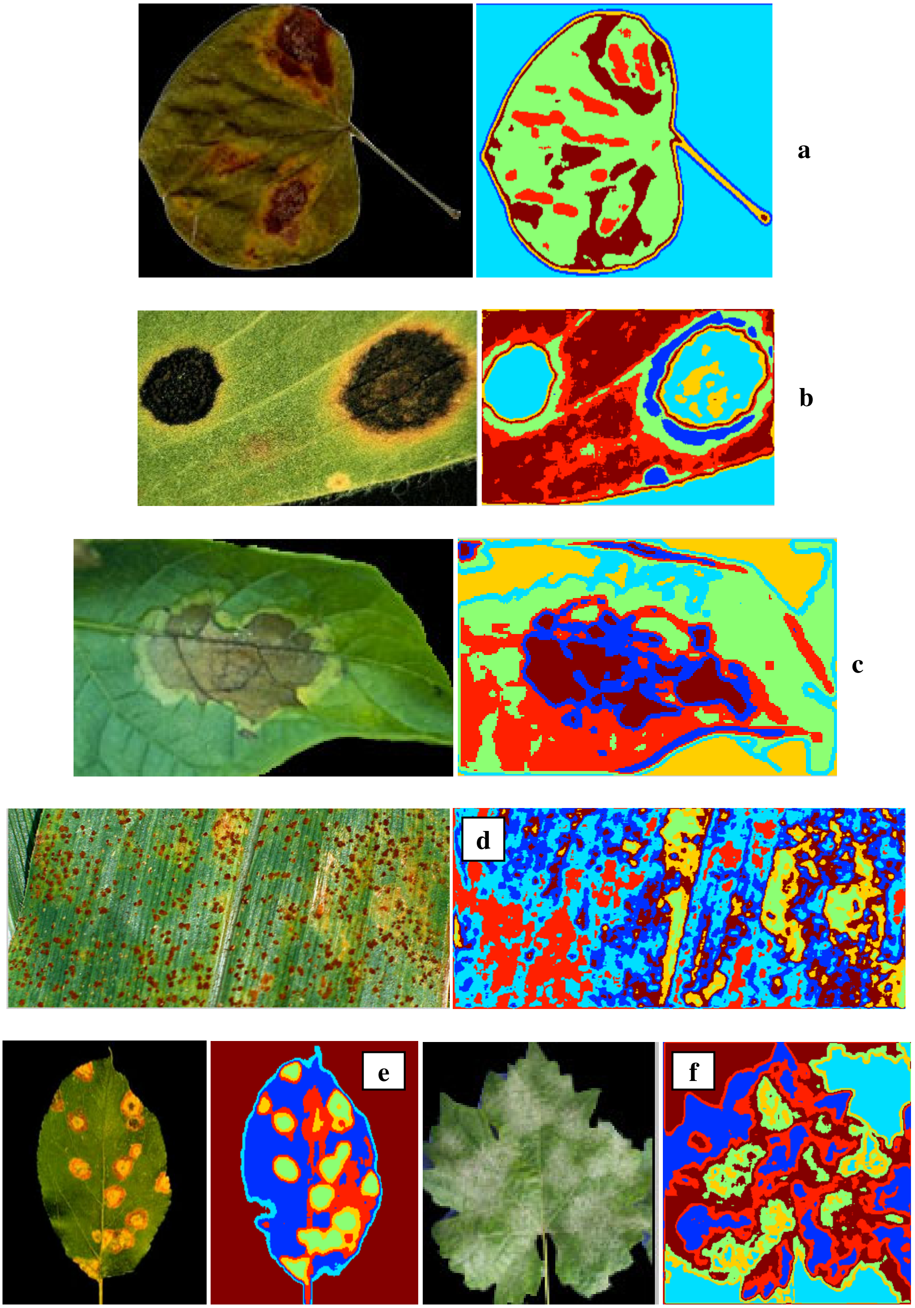


Fig. 6. Some of the outputs. a) Dark red sport. b) Tikka disease in groundnut leaf. c) Early Blight disease in tomato leaf. d) Rust disease in corn leaf. e) Rust disease in apple leaf. f) Powdery Mildew disease in graph leaf.

The proposed system works very well for the diseased leaf dataset and segment out the diseased area properly than the other methods. Thus, block-based unsupervised diseased plant leaf image segmentation is the best approach for detection of disease in plant leaf.

The complete system was designed in Matlab 7.7 (R2008b) software and also tested. It provides an easy platform for image processing and other mathematical operations. This paper is the continuation of [17], in which plant leaf species recognition based on leaf image was proposed. The some of the results are shown in the figure 6.

Table 1. Comparison of different methods

Approaches/Methods	**Real Life Image**	**Remark**
Histogram-Based Method	Average	Difficult to identify peaks and valley significant
Edge-Based Method	Below average	Disconnected edge
Proposed Method	***Good (better result)***	***Energy minimization***

5 Conclusion

This paper proposes new block-based unsupervised diseased plant leaf image segmentation method and observed that the results of this method is better than the already existing methods such as histogram-based segmentation and edge-based segmentation specially for the diseased leaf dataset. The advantage of this method is that it's very simple and easy to implement with higher accuracy. Also there is no need of fixing any threshold value for segmentation.

Experimentally it is found that this method is better than the other supervised segmentation.

The future work will include some other unsupervised segmentation methods for higher accuracy such as graph-cut method to overcome the limitation of this method.

References

1. Deng, Y., Manjunath, B.S.: Unsupervised Segmentation of Color-Texture Regions in Images and Video. IEEE Trans. Pattern Analysis and Machine Intelligence 23, 800–810 (2001)
2. David, J.F., Yau, K.Y., Elmagarmind, A.K., Aref, W.G.: Automatic Image Segmentation by Integrating Color-Edge Extraction and Seeded Region Growing. IEEE Trans. Image Processing 10, 1454–1466 (2001)
3. Chen, K.M., Chen, S.Y.: Color Texture Segmentation using feature Distributions. Pattern Recogntion Lett. 23, 755–771 (2002)
4. Glasbey, C.A.: An Analysis of Histogram-based Thresholding Algorithm. CVGIP: Graphical Model and Image Processing 55, 532–537 (1993)

5. Naemura, M., Fukuda, A., Mizutani, Y., Izumi, Y., Tanaka, Y., Enami, K.: Morphological Segmentation of Sport Scenes using Color Information. IEEE Transactions on Broadcasting 46(3) (September 2000)
6. Leing, T., Malik, L.: Contour Continuity in Region Based Image Segmentation. In: Proceedings of the Fifth Europe Conference on Computer Vision (June 1998)
7. Brejl, M., Sonka, M.: Edge-Based Image segmentation: Machine Learning from Examples. In: IEEE World Congress on Computational Intelligence, pp. 814–819 (May 1998)
8. Meyer, F.: Color image segmentation. In: Proc. Int. Conf. Image Processing, Maastricht, The Netherlands (1992)
9. Bhanu, B., Lee, S., Ming, J.: Adaptive image segmentation using a genetic algorithm. IEEE Transactions on Systems, Man, and Cybernetics 25(12) (December 1995)
10. Wang, S., Siskind, J.M.: Image segmentation with ratio cut. IEEE Transaction on Pattern Analysis and Machine Intelligence 25(6) (June 2003)
11. Yamazaki, T.: Introduction of EM Algorithm into Color Image segmentation. In: Proceedings of ICIRS 1998, pp. 368–371 (August 1998)
12. Dustmann, C.: Maximum Likelihood (ML) Estimation. Lectures 8/9, Spring Term (2003), `http://www.ucl.ac.uk/~uctpa36/lec8_03.pdf`
13. Wu, Y., Yang, X., Chan, K.L.: Unsupervised color image segmentation based on Gaussian mixture model. In: Proceedings of the Fourth International Conference on Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia, vol. 1, pp. 541–544 (2003)
14. Osher, S., Sethian, J.: Fronts propagating with curvature dependent speed: Algorithms based on the Hamilton–Jacobi formulation. Journal of Computational Physics 79, 12–49 (1988)
15. Osher, S., Paragios, N.: Geometric Level Set Methods in Imaging, Vision and Graphics. Springer, Heidelberg (2003)
16. Juan, O., Keriven, R.: Unsupervised Segmentation for Digital Matting. CERTIS Research Report November (2004), `http://certis.enpc.fr/publications/papers/05certis04.pdf` (July 30, 2011)
17. Prasad, S., Kudiri, K.M., Tripathi, R.C.: Relative Sub-image Features for Leaf Recognition using Support Vector Machine. In: Processing of International Conference on Communication Computing and Security, ICCCS 2011, pp. 343–346 (February 2011)

Measuring the Weight of Egg with Image Processing and ANFIS Model

Payam Javadikia[1], Mohammad Hadi Dehrouyeh[2], Leila Naderloo[3],
Hekmat Rabbani[1], and Ali Nejat Lorestani[1]

[1] Department of Mechanical Engineering of Agricultural Machinery,
Collage of Agriculture and Natural Resources, Razi University, Kermanshah, Iran
`{pjavadikia,ali.lorestani}@gmail.com, hrabani2000@yahoo.com`
[2] Department of Mechanical Engineering of Agricultural Machinery,
Roudehen Branch, Islamic Azad University, Roudehen, Iran
`dehrouyeh@yahoo.com`
[3] Department of Mechanical Engineering of Agricultural Machinery,
Faculty of Agricultural Engineering and Technology,
Collage of Agriculture and Natural Resources, University of Tehran, Tehran, Iran
`lnaderloo@gmail.com`

Abstract. It is clear that the egg is very important in human food basket. But a problem in food processing manufactures is measuring the weight of eggs as real time and it's difficult. One solution can be by using the camera. In this research we tried to measure the width and length of egg by real time image processing and then design and optimize an ANFIS model to find best relation between image processing outputs and the weight of egg. The correlation coefficient of experimental value for weight of egg and predicted value by ANFIS model is 0.9942. The result is very interesting and this idea is cheap, novel and practical.

Keywords: egg, weight, defect, grading, adaptive neuro-fuzzy inference system.

1 Introduction

In modern egg processing plants, the inspection and quality assessment of eggs defects is very important because it is largely done by human experts. This operation suffers from several drawbacks due mainly to human subjectivity, visual stress and tiredness. Defects searched for include bloodspots, cracks, and dirty shells. A bright light is directed through the eggs and the defects become more visible and easier to detect [1].

Research in the automation of the inspecting operation for eggs is based on two different techniques: (1) camera vision and image analysis [2] and (2) mechanical stiffness measurements. Image analysis by machine vision has been studied as a component of automatic egg sorting. The accuracy of the vision methods depends on the resolution of the camera, the sorting algorithm and the color of the eggshell [3].

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 407–416, 2011.
© Springer-Verlag Berlin Heidelberg 2011

Reference [4] presented a nondestructive prediction method for yolk/albumin ratio in chicken eggs by computer image analysis. Reference [5] used discontinuities determination by applying an adaptive threshold on the filtered image for detecting eggshell defects. Reference [6] used image processing method for development of algorithms for detection of internal blood spots and dirtiness of eggshell.

Egg geometrical calculations that include estimations of volume and surface area are important for the poultry industry and in biological studies, as they can be used in research on population and ecological morphology and to predict chick weight, egg hatchability, shell quality characteristics and egg interior parameters. Also weight of egg is one of the most important parameters for egg sorting [1].

Reference [7] calculated egg geometry using measurement of length and width of egg. The image processing technique is a useful method for the surface area and volume determination of objects. Reference [8] used machine vision to determine the surface area and the volume of axisymmetric agricultural products. Reference [9] used image processing to determine the volume of watermelons. Reference [10] estimated the volume of egg by image processing technique and spheroid approximation.

Artificial intelligent methods such as fuzzy logic (FL) and artificial neural network (ANN) have been used in many engineering processes [11]. Today the fuzzy logic offers the mathematical methods that allows for a simple knowledge representation of the production control principles in terms of IF-THEN rules [12]. Fuzzy logic allows obtaining defined solutions from vague, ambiguous or uncertain information similar to human mind and solving non-linear problems [13]. Both artificial neural network (ANN) and fuzzy logic (FL) are used in ANFIS structure [14]. The neuro-fuzzy is combinative of ANN and fuzzy system, has benefits of two models. Usage of the adaptive neuro-fuzzy inference system (ANFIS) is a useful method for the solution of function approximation problems and fast solutions [15, 16]. A neuro-fuzzy system is capable of modeling highly nonlinear and approximately known systems [17]. This method is significantly useful in the engineering performance where classical methods fail or they are too complex to be used [18].

Reference [19] carried out a study to detect defects in eggs by training neural networks with color histograms obtained from color images. There has been applications of grading eggs with BP (Back Propagation) Neural Network [20] and reached accuracy of 91%. In apples grading, the accuracy of 90.8% has been reported with machine vision and BP Neural Network [21]. Reference [22] applied BP Neural Network to classify the plants and reached accuracy of 90%. Broccoli grading was carried out by [23] based on computer vision and neural networks and accuracy of 93.4% was reached.

The object of this research is designing the method to measure the weight of egg as real time and the used idea is novel. In another words we measured the width and length of egg and then find the best ANFIS model to predict the weight via width and length of eggs (outputs of image processing).

2 Materials and Methods

2.1 Hardware

In this research, we used white eggs in different sizes from small to large in five categories. For size detection, 200 eggs were selected from a commercial packing station.

Figure 1 shows a view of used hardware system. The eggs were placed in the illumination box and the camera had been placed above it, took images from them. The wooden cubic illumination box with dimensions of 40×40×40 cm prepares possibility of light adjustment and light noises removing. A SAMSUNG™ CCD camera, SDN-550 model was used for imaging and the distance between lens and egg was approximately 200mm. The camera has 768 and 576 pixels in horizontal and vertical directions respectively and its resolution is 530 TV lines.

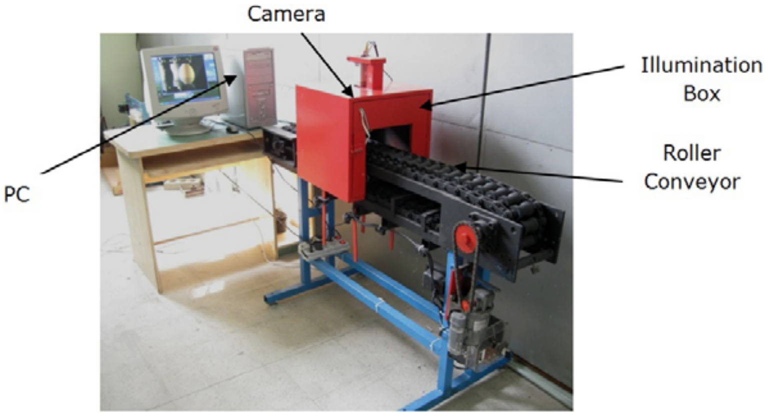


Fig. 1. The hardware system

The camera was connected to a personal computer (PC) through a WINFAST™ frame grabber. This device digitizes the images and saves them on the hard drive. The PC was used for observation and analysis of images. The illumination system is lighting from down of roller conveyor. In this case, a 50W halogen lamp was installed below of rollers of conveyor that was carrying the eggs.

2.2 Software

The overall performance of image processing algorithm to detect the size of the egg consists of three steps (Figure 2). The first step in image processing was the preprocessing operation. At first, the image was called by MATLAB software and then the Algorithm was used to process the image.

Figure 3-a, shows a picture of an egg for example. The dimensions of the image were converted to a square matrix. The processing time per image was decreased by removing unnecessary data of image significantly (Figure 3 -b).

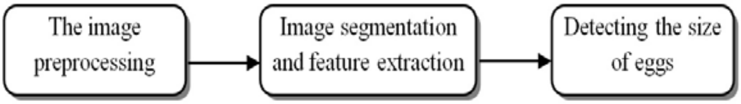

The image preprocessing
Image segmentation and feature extraction
Detecting the size of eggs

Fig. 2. The overall performance of the size detection algorithm

Next, the color image in RGB space was converted to HSV space (the principle of color, saturation and value) (Figure 3 -c). Also, Figure 3 –d shows gray scale image converted from the color image in RGB color space.

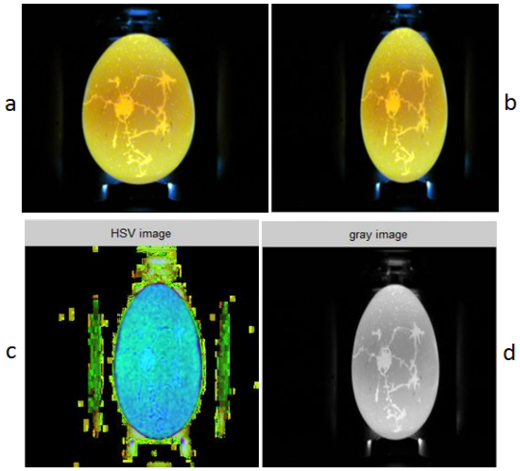


Fig. 3. a) original image, b) resized image, c) image in the HSV color space and d) grayscale image

Next step was the image segmentation. The result of this step was a color image belongs to only egg and other parts were pixels completely black with zero intensity. Figure 4 shows the segmented image of egg.

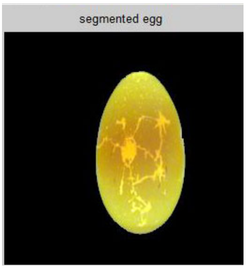


Fig. 4. The segmented image

The size of egg is an important factor in egg grading. Because of ease the weight measurement of egg, this factor is used to grading in egg production at manufactures while normally customers are paying more attention to the size of eggs. Two parameters were considered as criteria for this stage:

- Counting the pixels of egg image: In this method the area of eggshell can be obtained by counting the pixels in the eggshell image. Uniformity images can be achieved by fixing the camera situation, not changing the distance between the camera and eggs and not changing the zoom lens of the camera. So there is no need for calibration images and it is enough only to obtain a good correlation between the number of pixels in image and real size of egg.

Big and small diameter size of egg: The exterior shape of the egg image is very similar to the oval shape. Big and small diameter size of egg instead of big and small diameter of the oval can specify the area of the eggshell in the image approximately. Also after measurement of the big and small diameters of the egg and dividing them to each other, we can achieve to a factor that is known as golden ratio. This parameter shows the exterior proportion of egg that it can be considered in the market of product.

Before The final image of previous step (the egg image separated from background) is an image in RGB color space. The red color surface (R) was extracted and its non-zero values that was defined as a separate image. Figure 5-a, shows this picture. In many case, the continuous regions on the surface of the eggshell at resulting image remain as non-homogenous mass with eggshell. To fix this problem, the imfill function was used. The algorithm of this function is based on morphological reconstruction of on binary images [3]. Figure 5 shows the image after using of the imfill function. In final step, non-zero pixels were counted and output number recognized the pixels number of eggshell.

To determine the big and small diameter size of egg, the edges should be detected in eggs. To do this, the canny method of edge detector was selected with edge function and the last image of previous step (counting the number of pixels) was used. Figure 5 shows the result of using the edge function with 0.7 thresholds. The big and small diameters of the egg are also visible in this picture.

In the matrix of resulting image, the rows and columns with non-zero values (the border edge of the egg) were arranged in descending order in the original matrix depend on their position. The minimum and maximum values of resulting matrix (in both x and y directions) denote the four points shown in Fig 5. Finally, the big and small diameters of egg was obtained in terms of the number of pixels by subtraction of minimum and maximum values from each other in both x and y directions.

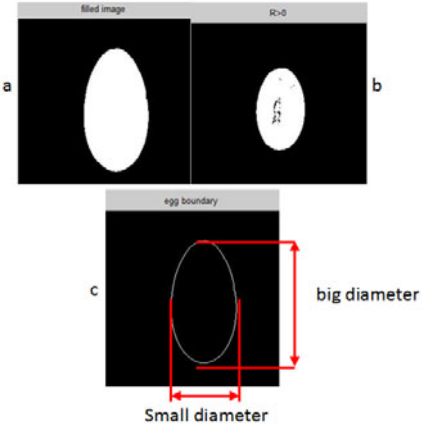


Fig. 5. a) red color surface b) the picture filled by the imfill function c) identify the edge of the egg

2.3 Adaptive Neuro-Fuzzy Inference System (ANFIS)

The two learning methods are used in an ANFIS to give the mapping relation between the input and output data and to determine the optimal distribution of membership functions [24]. These learning methods are back propagation and hybrid that hybrid system is combination of back propagation and least squares type method [24, 25, and 26]. In back propagation method, the gradient descent algorithm update the weight parameters [15].The parameters related with the membership functions will modify through the learning process. Gradient vector facilitate the computation of these parameters. Once the gradient vector is obtained, any of the several optimization routines could be applied to adjust the parameters and to reduce error [27, 13].

The adaptive neuro-fuzzy inference system (ANFIS) is based on Sugeno-type system and can specify the relation between input and output data [15, 28]. It is mainly based on the fuzzy 'if-then' rules from the Takagi and Sugeno type which is shown in Figure 6 [28]. It includes a premise part and consequent part. The equivalent ANFIS structure of Takagi and Sugeno form is shown in Figure 7. Two inputs (x and y) and one output (f) have been spotted in the fuzzy inference system, to simplify the ANFIS procedure explanation. The ANFIS comprises five layers in this inference system. Each layer involves several nodes, which are described by the node function. The input signals in the present layer have been obtained from output signals of nodes in the previous layers. The output will be served as input signals for the next layer after manipulation by the node function in the present layer. Here, square nodes, named adaptive nodes, are adopted to show that the parameter sets in these nodes are adjustable [13].

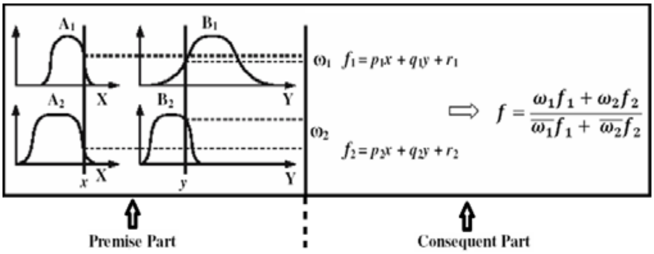


Fig. 6. Sugeno fuzzy 'if-then' rule and fuzzy reasoning mechanism

Where, circle nodes are adopted to point that the parameter sets are fixed in the system and to this they named fixed nodes. With regard to the number of input and output, the rule base will contain two fuzzy 'if-then' rules as follows [24, 25, 26, and 13]:

Rule 1: if x is A1 and y is B1 then F = p1x+ q1y+r1.
Rule 2: if x is A2 and y is B2 then F = p2x+ q2y+r2.

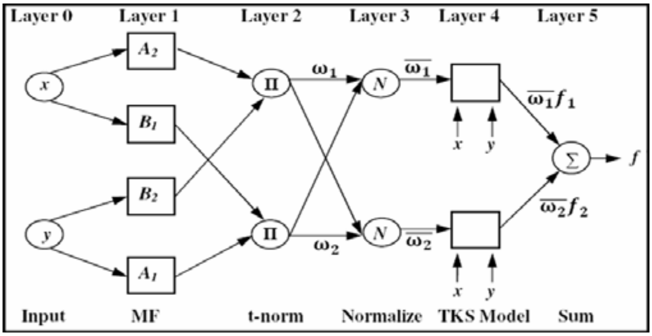


Fig. 7. The structure of ANFIS

3 Results and Discussion

As it’s shown in Figure 8, in this research we finally found the best model to predict the output (weight of egg) that has two inputs that are: the number of width pixel and length pixel of egg. We changed five important parameters of ANFIS Network to find best ANFIS model. These parameters are: number of membership functions for each input, type of membership function (trimf, trapmf, gbellmf, gaussmf, gauss2mf, primf, desigmf, and psigmf), type of output membership function (constant or linear), optimization method (hybrid, back propackation) and number of epochs. So we changed all of them and compared the error of each ANFIS with together and selected the best ANFIS model with minimum error. The properties of best ANFIS model were: 7 membership functions for each input and from “gbellmf” type and linear type membership function for output and hybrid method for optimization with 100 epochs. Then we calculated some other parameters of the best model performance such as MSE (Mean Square Error), MAE (Mean Absolute Error), SSE (Sum Square Error), R (Correlation Coefficient) and P (probability) that respectively were: 0.2955, 0.3285, 35.4649, 0.9942 and 0.

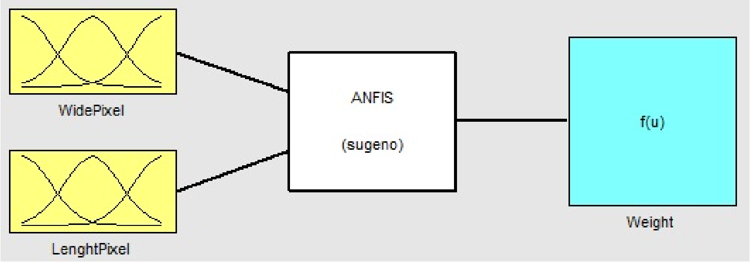


Fig. 8. The number of inputs and output in best model

For example in figure 9, influence of change in number of pixels in width and length egg on the weight of egg is shown. In left picture the result of ANFIS model without filtering is shown and the right picture is the result of model with filtering the out of range results. ANFIS method can simulate the result of data that are not in experimental database and when this simulation is true that the experimental database covers all of range. Here as you can see in left picture the actual model is invisible because of small scale, in another words the simulated results of ANFIS for out of range data of experimental database have big value and so the actual model that have small value in weight results is invisible. To solve this problem, a filter that clean the big value in the weight of eggs is used and therefore the graph of model is result in right picture of figure 9.

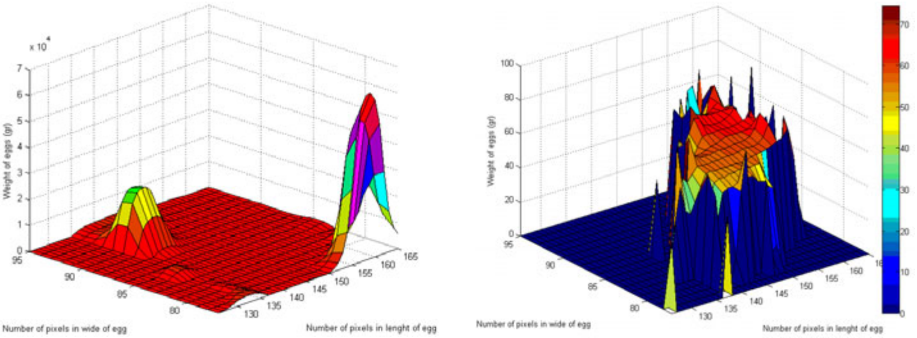


Fig. 9. Relation of number of pixels in width and length of egg influence on weight

Finally for testing the performance of ANFIS model we draw the graph of experimental and predict values in figure 10, as seen in this graph, the predicted value by ANFIS model was equal to experimental value approximately.

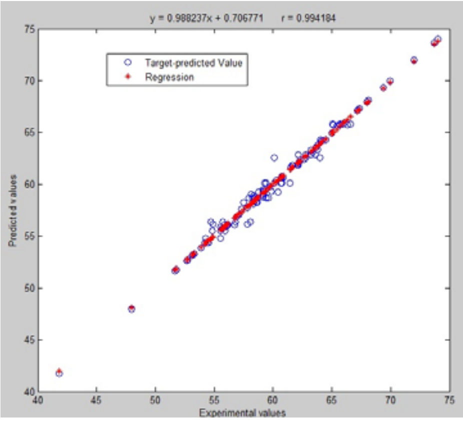


Fig. 10. Experimental value and predict value of ANFIS model

Rashidi and Gholami (2011) predicted the egg mass based on some geometrical attributes. They suggested nine linear regression models for predicting egg mass based on length, diameter, geometrical mean diameter, first projected area, second projected area, criteria area and estimated Volume. They results showed that the mass model based on geometrical mean diameter had $R^2 = 0.595$ and the mass model based on length and diameter had $R^2 = 0.619$. In addition, the mass model based on the first projected areas with $R^2=0.599$. But in this study, the ANFIS model had two inputs (number of pixels in width and length of egg) to predict the weight of egg and R^2 obtained as 0.98.

4 Conclusion

In this research we tried to find the suitable model to predict the weight of eggs by image processing that is very important in food processing manufactures. Anyway after calculating the length and width of egg via image processing we tried to find any significant relation between them in relate to weight of egg and finally we find the best model by ANFIS methodology that can predict the weight of egg by 0.99 correlation coefficient value.

References

1. United States Department of Agriculture. Egg-Grading Manual. Agriculture Handbook No.75. Agriculture Marketing Service, USDA (2000)
2. Goodrum, J.W., Elster, R.T.: Machine Vision for Crack Detection in Rotating Eggs. Transactions of the ASAE 35(4), 1323–1328 (1992)
3. Gonzalez, R.C., Woods, R.E.: Digital Image Processing, 2nd edn. Prentice Hall Inc. (2004)
4. Kuchida, K., Fukaya, M., Miyoshi, S., Suzuki, M., Tsuruta, S.: Nondestructive Prediction Method for Yolk: Albumin Ratio in Chicken Eggs by Computer Image Analysis. Poultry Science 78, 909–913 (2000)
5. Pourreza, H.R., Pourreza, R.S., Fazeli, S., Taghizadeh, B.: Automatic Detection of Eggshell Defects Based on Machine Vision. Journal of Animal and Veterinary Advances 7(10), 1200–1203 (2008)
6. Dehrouyeh, M.H., Omid, M., Ahmadi, H., Mohtasebi, S.S., Jamzad, M.: Grading and Quality Inspection of Defected Eggs Using Machine Vision. International Journal of Advanced Science and Technology 17, 23–31 (2010)
7. Narushin, V.G.: Egg Geometry Calculation Using the Measurements of Length and Breadth. Poultry Science 84, 482–484 (2005)
8. Sabliov, C.M., Bolder, D., Keener, K.M., Farkas, B.E.: Image Processing Method to Determine Surface Area and Volume of Axisymmetric Agricultural Products. International Journal of Food Prop. 5, 641–653 (2002)
9. Koc, B.: Determination of Watermelon Volume Using Ellipsoid Approximation and Image Processing. Journal of Postharvest Technology 45, 366–371 (2007)
10. Rashidi, M., Malekiyan, M., Gholami, M.: Egg Volume Determination by Spheroid Approximation and Image Processing. World Applied Sciences Journal 3(4), 590–596 (2008)

11. Fraile-Ardanuy, J., Zufiria, P.J.: Design and comparison of adaptive power system stabilizers based on neural fuzzy networks and genetic algorithms. Neurocomputing 70, 2902–2912 (2007)
12. Kiralakis, L., Tsourveloudis, N.C.: Modeling and Optimization of Olive Stone Drying process. In: WSEAS International Conference on Dynamical Systems and Control, Venice, Italy, November 2-4, pp. 240–246 (2005)
13. Singh, J., Singh, S.: Modelling for tensile strength of friction welded aluminium pipes by ANFIS. Int. J. Intelligent Engineering Informatics 1(1), 3–20 (2010)
14. Avci, E.: Comparison of wavelet families for texture classification by using wavelet packet entropy adaptive network based fuzzy inference system. Applied Soft Computing 8, 225–231 (2008)
15. Buragohain, M., Mahanta, C.: A novel approach for ANFIS modelling based on full factorial design. Applied Soft Computing 8, 609–625 (2008)
16. Ross, T.J.: Fuzzy logic with engineering applications, 2nd edn. John Wiley and Sons Ltd., London (2004)
17. Cheng, C.B., Cheng, C.J., Lee, E.S.: Neuro-Fuzzy and Genetic Algorithm in Multiple Response Optimization. Computers and Mathematics with Applications 44, 1503–1514 (2002)
18. Ertunc, M., Hosoz, H.: Comparative analysis of an evaporative condenser using artificial neural network and adaptive neuro-fuzzy inference system. International Journal of Refrigeration 3(1), 1426–1436 (2008)
19. Patel, V.C., Mc Clendon, R.W., Goodrum, W.: Color Computer Vision and Artificial Neural Networks for the Detection of Defects in Poultry Eggs. Artificial Intelligence Review 12, 163–176 (1998)
20. Wang, Q.-H., Wen, Y.-X.: Research on the Grading Method of Egg's Weight Based on BP Neural Network. Hubei Agricultural Sciences (1), 97–99 (2005)
21. Long, M.-S., He, D.-J., Ning, J.-F.: An integrated apple grading system based on genetic neural network. J. Northwest Sci-Tech University of Agriculture and Foresty 12(6), 1–4 (2001)
22. Ma, H.M.: Application of BP neural networks in plant taxonomy identification. Agriculture Network Information (12), 1–3 (2006)
23. Tu, K., Ren, K., Pan, L., Li, H.: A Study of Broccoli Grading System Based on Machine Vision and Neural Networks. In: Proceedings of the 2007 IEEE International Conference on Mechatronics and Automation, Harbin, China, August 5-8 (2007)
24. Ying, L.C., Pan, M.C.: Using adaptive network based fuzzy inference system to forecast regional electricity loads. Energy Conversation and Management 49, 205–211 (2008)
25. Sengur, A.: Wavelet transform and adaptive neuro-fuzzy inference system for color texture classification. Expert Systems with Applications 34, 2120–2128 (2008)
26. Übeyli, E.D.: Adaptive neuro-fuzzy inference system employing wavelet coefficients for detection of ophthalmic arterial disorders. Expert Systems with Applications 34, 2201–2209 (2008)
27. Singh, J., Singh, S.: Multi input single output fuzzy model to predict tensile strength of radial friction welded GI pipes. International Journal of Information and Systems Sciences, Institute for Scientific Computing and Information 4(3), 462–477 (2008)
28. Jang, J.S.R.: ANFIS: adaptive-network-based fuzzy inference system. IEEE Trans. Syst. Man. Cybern., 665–685 (1993)
29. Rashidi, M., Gholami, M.: Prediction of egg mass based on geometrical attributes. Agriculture and Biology Journal of North America 2(4), 638–644 (2011)

Palmprint Authentication Using Pattern Classification Techniques

Amioy Kumar, Mayank Bhargava, Rohan Gupta, and Bijaya Ketan Panigrahi

Department of Electrical Engineering, Indian Institute of Technology Delhi, Hauz Khas,
New Delhi, India-110016

Abstract. Biometric technology incorporates several physiological and behavioral traits for personal authentication whenever deployed for security systems. Palmprint is one of the physiological trait has been utilized several times for key applications. This paper proposes a pattern classification approach for palm print authentication which utilizes soft computing techniques to classify a claimed identity into its appropriate class. The presented approach operates on feature level classification using 2D Gabor filter for feature representation and Principal Component Analysis (PCA) for computing weights as features .These features are used to train the classifiers by taking each user as a separate class. K-Nearest Neighbor, (KNN) and Probabilistic Neural Network (PNN) based classifiers are utilized in classification. These classifiers are also employed for score level classification by computing the matching scores using normalized hamming distance. The experiments are carried out on HongKong PolyU database which has been a benchmark database for palmprint authentication. The proposed techniques operate on very low false acceptance rate (FAR) of .0011% and false rejection rate (FRR) of 3% which shows the reliability of the proposed work.

Index Terms: Biometric Authentication, Palmprint, PCA, *K*-NN, PNN, Gabor filter.

1 Introduction

Biometrics accurately identifies or verifies individuals based upon each person's unique physical or behavioral characteristics. Leading examples are biometric technologies that recognize and authenticate faces, hands, fingers, signatures, irises, voices, and fingerprints. However, hand-based biometrics, like fingerprint, palmprint, hand geometry, hand veins etc., has attracted an increasing amount of attention due to their reliability and high user convenience. In the recent decade, palmprint is recognized as one of the most reliable biometric having features like, principal lines, moderate sized wrinkles, thin ridges and minutiae [1]. These are permanent in nature unless physically damaged [2]. Moreover, discriminative principal lines and wrinkles can be extracted even from low resolution images [1]. All these factors make the palmprint as an important biometric trait, especially, for low cost and medium security applications [2].

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 417–424, 2011.
© Springer-Verlag Berlin Heidelberg 2011

In the biometric literature, several palmprint authentication systems are designed using distinct feature representations, matching methods, and classification techniques [4]-[6],[9]-[11].However, most of these efforts are focused on ROC (Receiver Operating Characteristic) based selection of decision threshold by computing the matching scores from the enrolled users [4]. A pattern classification based approach [11] which utilizes a classifier to automatically choose a decision threshold from the trained samples, is still a less researched problem. The enrolled users are divided into genuine and imposter classes. While the genuine class is meant for acceptance imposter is for rejection. We utilize two approaches for classification: first, using features extracted by Gabor filter [4] and weighted features using PCA, second using genuine and imposter matching scores computed from these features.

In this paper, our goal is to use pattern classification approaches in the palmprint authentication in order to achieve better authentication results. The block diagram of the complete system is shown in Fig. 1.

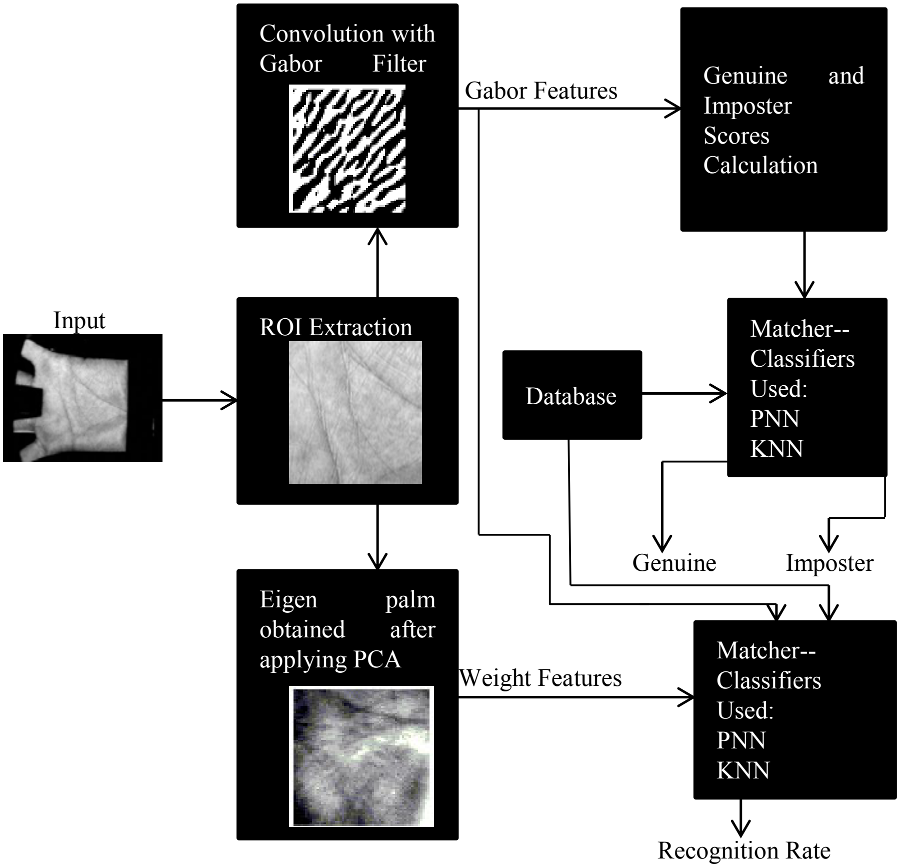

Convolution with Gabor Filter
Gabor Features
Genuine and Imposter Scores Calculation
Input
ROI Extraction
Matcher-- Classifiers Used: PNN KNN
Database
Genuine
Imposter
Eigen palm obtained after applying PCA
Weight Features
Matcher-- Classifiers Used: PNN KNN
Recognition Rate

Fig. 1. Block Diagram of Palmprint Authentication system

The rest of this paper is organized as follows: Gabor filter based feature extraction and PCA weight computation is discussed in Section 2. The experimental results are carried out in Section 3, and finally the conclusions are drawn in section 4.

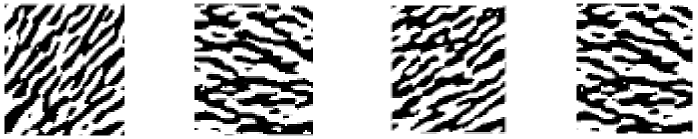
2 Feature Extraction

2.1 Gabor Filter

The Gabor filter has been utilized in palmprint [4], iris [6] and fingerprint for textural feature representation. In this paper, texture features from palmprint images are extracted applying 2-D Gabor filter. The circular Gabor filter is an effective tool for texture analysis [6], and has the following general form;

$$G(x,y,\theta,u,\sigma)=\frac{1}{2\pi\sigma^2}exp\left\{-\frac{x^2+y^2}{2\sigma^2}\right\}\exp\left(2\pi i(uxcos\theta+uysin\theta)\right) \tag{1}$$

Where $i=\sqrt{-1}$, u is the frequency of the sinusoidal wave, , θcontrols the orientation of the Function, and σ is the standard deviation of the Gaussian envelope. The Gabor filter is turned to zero DC to make it more robust against brightness [4]. The adjusted Gabor filter is used to filter the preprocessed images. Size of filter was taken to be 17×17.



(a) $\theta = 45°$ (b) $\theta = 0°$ (c) $\theta = 22.5$ (d) $\theta = 180°$

Fig. 2. Gabor Filter implemented in different orientations

2.2 Principal Component Analysis (PCA)

PCA is a useful statistical technique and broadly utilized for finding uncorrelated patterns in the input data [12]. Once the uncorrelated patterns are separated in the data, it can be compressed without much loss of information. In this paper, PCA is utilized to calculate weights from palm images which are further utilized as feature vector in pattern classification. This point is the key which helps us to reduce the feature matrix size. We apply PCA to obtain a set of basis images (Eigenpalms) which capture the inherent structure of palmprint texture [4]. We use these eigenpalms to calculate weights which are treated as features and thus helping to reduce the size of feature matrix. The following algorithm is used:

1. Subtract the mean from given image dataset.
2. Calculate the covariance matrix.
3. Calculate the eigenvectors and eigenvalues of the covariance matrix.

4. Now we calculate new data set (eigenpalms) by multiplying the eigenvector matrix (with eigenvector in rows) with the Mean adjust data matrix obtained after step 1. Applying whitening transform that is, making the rows uncorrelated and their variances equal to unity, often makes further processing easier. Mathematically speaking whitening transform $\mathrm{V1} = \mathrm{D}^{-(.5)}\mathrm{V}'\mathrm{K}$; where D denotes the eigenvalues on the diagonal V denotes the eigenvector matrix and K is the zero mean matrixes of N images. By applying D^{-1}instead of $\mathrm{D}^{-(.5)}$ we were able to improve our results further.
5. Now we select M eigenpalms corresponding to M highest eigenvalues.
6. Now we obtain a weight matrix by multiplying the eigenpalms obtained in step 5 with the mean adjust data set transposed.

The weights obtained basically shows the contribution of each Eigenpalm in representing the input palm image [5].

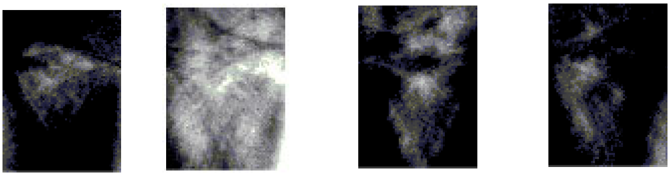

(a) (b) (c) (d)

Fig. 3. Some basis palmprint images obtained after applying PCA

3 Experimental Results

The palmprint is emerging as a promising biometric trait having several advantages over other biometric traits like the fingerprint. These include: (1) large surface area: ROI of the palm print is larger than that of the fingerprint and hence more information is available for processing. (2) The chances of palm getting damaged are very rare. (3) The presence of dirt or grease does affect the performance a bit. The palmprint database utilized in this work is from Hong Kong PolyU V2.

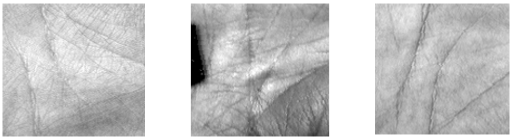

(a) (b) (c)

Fig. 4. Sample ROI used in database

The ROI and feature extraction methods are taken from [4]. The palmprint images of size 384×384 are utilized to extract ROIs of size 128×128. Sample palmprint images from database are shown in Fig. 4.

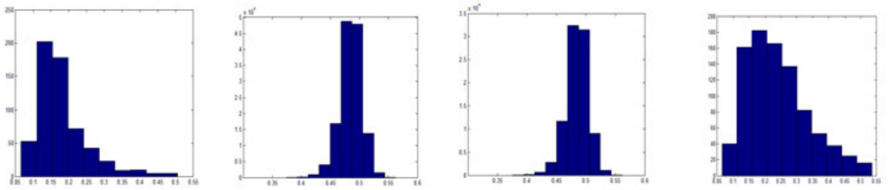
3.1 Verification Rate

Normalized hamming distance has been used as a criterion for classifying a user as genuine or imposter [4]. A normalized Hamming distance used in [6] is adopted to determine the similarity measurement for palmprint matching. Let A and B are two palmprint images then normalized hamming distance is given by

$$D = \frac{\mathrm{xor(A,B)}}{\mathrm{N*N}} \text{where } \mathrm{N} * \mathrm{N} \text{ denotes the size of image} \tag{2}$$

We have calculated hamming distance on the images obtained after application of Gabor filter .In PolyU database version 1 of 100 users we have 5 sample images of each user. Among them 3 were taken for training and rest 2 for testing. Now each image in the training database is having 3 genuine scores so we have a total of 300 genuine scores. Similarly each image has 99 imposter images so total of 29700 imposter scores. Now each of the images in testing database is compared to its 3 corresponding images in training database to create a total of 200 genuine scores and 18000 imposter scores.

By using PNN (Probabilistic neural network) for classification on the basis of above scores, we were able to achieve a FRR of 3.5% with a FAR of 0% on PolyU Database version 1 .In doing so the training genuine and imposter scores were divided into 2 classes and were fed to PNN. And when the testing scores were fed to the network the output showed the classes of Testing scores from which we calculated the FRR and FAR. On applying this technique to PolyU Database of 150 users (10 images of each user out of which 6 images were used in training set and 4 were used in testing set) we were able to achieve a FRR of 3 with a FAR of .0011%. There is no need to set any kind of threshold .The classifier automatically classifies the given input.



(a) (b) (c) (d)

Fig. 5. Distribution of (a) Genuine testing scores (b) Imposter Training scores (c) Imposter testing scores (d) Genuine training scores. Database used (PolyU database of 1500 images of 150 users.

3.2 Identification Rate

Recognition rates are calculated by classification using KNN and PNN. As mentioned earlier weights obtained on application of PCA and Gabor filter output are taken as features for recognition. The feature matrix obtained after application of Gabor filter was converted to binary before classification.

3.2.1 KNN

K-Nearest Neighbor (KNN) classification is a very simple, yet powerful classification method. The key idea behind KNN classification is that similar observations belong to similar classes. Thus, one simply has to look for the class designators of a certain number of the nearest neighbors and weigh their class numbers to assign a class number to the unknown. The weighing scheme was chosen to be majority based. The number of the nearest neighbors, k, should be odd in order to avoid ties. Various distance types are used in KNN Classification. [7]. Verification of a query palmprint is determined by the class of its k-nearest neighbor. If Class of a query palmprint image is same as the output of classifier, then palmprint is matched otherwise not matched.

To validate our verification algorithms we have calculated the Recognition Rate. For PolyU database version1 (500 images) 3 images of each user is used for training the KNN Classifier and remaining 2 for testing. Similarly in PolyU database version 2(4000 images) we have taken 6 images of one user for training and remaining 4 for testing.

Table 1. KNN applied on PolyU database version 1

Distance type	Feature length =40			Feature length =40 (without applying whitening transform)			Feature length =100		
	K=1	K=2	K=3	K=1	K=2	K=3	K=1	K=2	K=3
Cityblock	100	100	98.5	99.5	99.5	98.5	99.5	99.5	98.5
Correlation	99.5	99.5	98.5	96	96	94.5	99.5	99.5	98
Cosine	99.5	99.5	98.5	96	96	94.5	99.5	99.5	98
Euclidean	100	100	99	96.5	96.5	94.5	99	99	99

On the PolyU database version2 we achieved a recognition rate of 99.25% for which we chose Cityblock as our distance criterion and 40 as the length of features. The process of verification is very fast.

Table 2. KNN on Gabor outputs (PolyU Database version 1)

Hamming(k=2)	98%
Euclidean	96.5%
Cosine	97.5%
Cityblock	97%

3.2.2 PNN

Probabilistic neural networks (PNN) are a kind of radial basis network suitable for classification problems. PNN can be generalized well on the data that it has not seen before. Their design is straightforward and does not depend on training. A PNN is guaranteed to converge to a Bayesian classifier providing it is given enough training data. [8]

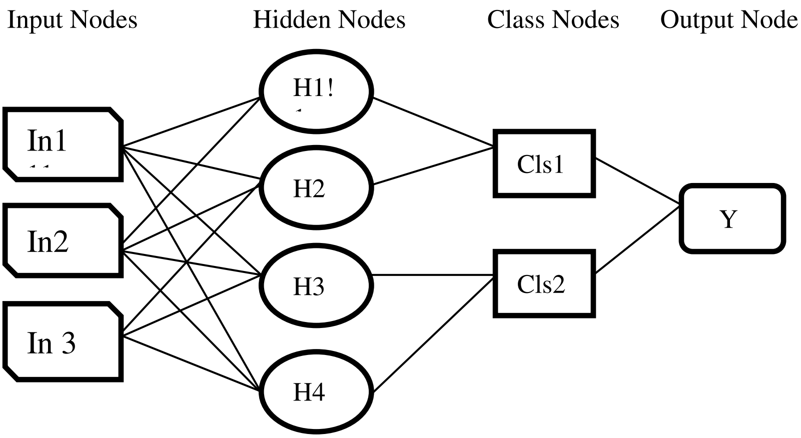


Fig. 6. Architecture of PNN

On Hong Kong PolyU database version 1 we were able to achieve a recognition rate of 98.5 % using PNN when we used PCA for feature extraction and chose the length of feature matrix as 27. We also implemented PCA analysis on the Gabor outputs and were able to achieve a recognition rate of 98.5 % after classification by PNN.

4 Conclusion

This paper presents a pattern classification approach for palmprint based biometric authentication. The usage of small size feature vectors results in reduced computational complexity, which is critical for personal recognition. By applying PCA to calculate weights reduced the feature matrix size and we were able to achieve better results and in lesser time. While classifying on the basis of scores PNN was able to achieve good results without having a need to set a threshold manually. KNN classifier resulted in better recognition rate as compared to PNN although PNN worked better while finding out the verification rate reflecting that when classes are less and training data is more PNN is able to classify more accurately than KNN . Classifier fusion could be used to improve the results further. Also score level calculation could be done on the weights which were obtained after applying PCA.

References

[1] Shu, W., Zhang, D.: Automated personal identification by palmprint. Optical Engineering 37(8), 2359–2362 (1998)
[2] Zhang, D.: Palmprint Authentication. Kluwer Academic Publishers (2004)

[3] PolyU Palmprint Database, Biometric Research Centre, Hong Kong Polytechnic University, `http://www.comp.polyu.edu.hk/~biometrics/`

[4] Zhang, D., Kong, W.K., You, J., Wong, M.: On-line palmprint identification. IEEE Trans. on PAMI 25(9), 1041–1050 (2003)

[5] Connie, T., Teoh, A., Goh, M., Ngo, D.: Palmprint recognition with PCA and ICA. In: Image and Vision Computing, Palmerston North, New Zealand, pp. 227–232 (November 2003)

[6] Daugman, J.G.: High confidence visual recognition of persons by a test of statistical independence. IEEE Transactions on Pattern Analysis and Machine Intelligence 15(11), 1148–1161 (1993)

[7] Song, Y., Huang, J., Zhou, D., Zha, H., Giles, C.L.: IKNN: Informative K-Nearest Neighbor Pattern Classification. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) PKDD 2007. LNCS (LNAI), vol. 4702, pp. 248–264. Springer, Heidelberg (2007)

[8] Specht, D.F.: Probabilistic Neural Networks. Neural Networks 3, 109–118 (1990)

[9] Kong, A., Zhang, D., Kamel, M.: A survey of palmprint recognition. Pattern Recognition 42(7) (2009)

[10] Lu, G., Zhang, D., Wang, K.: Palmprint recognition using eigenpalms features. Pattern Recognition Letters 24, 1463–1467 (2003)

[11] Kumar, A., Zhang, D.: Personal authentication using multiple palmprint representation. Pattern Recognition 38, 1695–1704 (2005)

[12] Smith, L.I.: A tutorial on Principal Components Analysis (February 26, 2002)

A Supervised Approach for Gene Mention Detection

Sriparna Saha[1,⋆], Asif Ekbal[1,⋆], and Sanchita Saha[2]

[1] Department of Computer Science and Engineering,
Indian Institute of Technology Patna, India
`{sriparna,asif}@iitp.ac.in`
[2] Haldia Institute of Technology, India
`sanchita.cse2007@gmail.com`

Abstract. Named Entity Recognition and Classification (NERC) is one of the most fundamental and important tasks in biomedical information extraction. Gene mention detection is concerned with the named entity (NE) extraction of gene and gene product mentions in text. Several different approaches have emerged but most of these state-of-the-art approaches suggest that individual NERC system may not cover entity representations with arbitrary set of features and cannot achieve best performance. In this paper, we propose a voted approach for gene mention detection. We use support vector machine (SVM) as the underlying classification methodology, and build different models of it depending upon the various representations of the set of features. One most important criterion of these features is that these are *identified and selected largely without using any domain knowledge*. Evaluation results with the benchmark dataset of GENTAG yields the state-of-the-art performance with the overall recall, precision and F-measure values of 94.95%, 94.32%, and 94.63%, respectively.

1 Introduction

The biological research literature is one of the vital repository of knowledge [2]. MedLine, the primary research database serving the biomedical community, currently contains over 14 million abstracts, with 60,000 new abstracts appearing each month. All of these resources are largely annotated manually, and the costs involved are huge. Thus the amount of information often prohibits updating previously annotated material to conform to changing annotation guidelines. This situation has naturally created to an interest in automated techniques for problems such as topic classification, word sense disambiguation, and tokenization in the biomedical domain (cf. MedLine's Indexing Initiative [1]).

Among few other evaluation challenges, BioCreAtIvE [1] gained significant attention to the community during the last few years. The goal of the BioCreAtIvE challenge was to provide a set of common evaluation tasks to assess the state-of-the-art for text mining applied to biological problems. Extracting gene names is not an easy task as these have complex structures. Millions of ambiguous genes exist and new genes are created

---

[⋆] First two authors are joint first authors.

[1] http://biocreative.sourceforge.net/

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 425–432, 2011.
© Springer-Verlag Berlin Heidelberg 2011

at the polynomial rate. Applying information extraction methodologies in the biomedical domain has become a growing research trend over the years. Gene (or, protein) name identification is the first step towards achieving good accuracies in information extraction.

The first edition of BioCreative was held in the year 2003 and the workshop took place in 2004. The 15 participants were given the training as well as testing corpora. The highest achieved F score for the gene mention detection was 82.2%. The second BioCreative challenge (BioCreative-II)[2] is relatively a recent competition for biological literature mining systems, and took place in 2006 followed by a workshop in April, 2007. This challenge consisted of a gene mention task, a gene normalization task and protein-protein interaction tasks. The highest accuracy for the gene mention task was reported to be 87.21% [3]. The BioCreative challenges are ongoing, and more and more teams are showing their interests into it.

In this paper we focus on the detection of gene mentions from the biomedical texts. This problem may be thought of as a Named Entity Recognition (NER) problem which requires the identification of names corresponding to shallow semantic categories. NER is an important component for more complex information extraction tasks such as automatic extraction of protein-protein interaction information. In this paper, we propose a voted approach for gene mention detection. Support Vector Machine (SVM) [7] is used as the underlying classification technique. Initially, we generate different models by varying the available features. We identify a very rich and effective feature set that includes variety of features based on orthography and local contextual information. One most important characteristic of our system is that the *identification and selection of features are mostly done without any domain knowledge and/or resources*. Thereafter we combine various models using majority and weighted voting approaches. We evaluate our approach on GENETAG[3] corpus which is an extended version of Biocreative-II data set. Evaluation results yield the overall recall, precision and F-measure values of 94.95%, 94.32% and 94.63%, respectively. Comparative evaluation results show that our proposed approach performs better than all the previous state-of-the-art systems that were developed using the same datasets.

2 Proposed Approach

We propose a voted approach for gene mention detection. We use SVM as the underlying classification methodology and generated different versions of it by considering various subsets of features.

2.1 Support Vector Machine Framework for Gene Mention Detection

Support Vector Machines (SVMs) [7] are applied to text categorization, and are reported to have achieved high accuracy without falling into over-fitting even though with a large number of words taken as the features [4,5]. Suppose, we have a set of training data for

---

[2] http://biocreative.sourceforge.net/

[3] ftp://ftp.ncbi.nlm.nih.gov/pub/tanabe/GENEATG.tar.gz

a two-class problem: $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$, where $\mathbf{x}_i \in R^D$ is a feature vector of the i-th sample in the training data and $y \in \{+1, -1\}$ is the class to which $\mathbf{x}_i$ belongs. In their basic form, a SVM learns a linear hyperplane that separates the set of positive examples from the set of negative examples with *maximal margin* (the margin is defined as the distance of the hyperplane to the nearest of the positive and negative examples). In basic SVMs framework, we try to separate the positive and negative examples by the hyperplane written as:

$$(\mathbf{w}.\mathbf{x}) + b = 0 \quad \mathbf{w} \in \mathbf{R}^n, b \in \mathbf{R}.$$

SVMs find the "optimal" hyperplane (optimal parameter $\overline{w}, b$) which separates the training data into two classes precisely.

The linear separator is defined by two elements: a weight vector $\mathbf{w}$ (with one component for each feature), and a bias b which stands for the distance of the hyperplane to the origin. The classification rule of a SVM is:

$$sgn(f(\mathbf{x}, \mathbf{w}, b)) \tag{1}$$

$$f(\mathbf{x}, \mathbf{w}, b) =< \mathbf{w}.\mathbf{x} > +b \tag{2}$$

being $\mathbf{x}$ the example to be classified. In the linearly separable case, learning the maximal margin hyperplane $(\mathbf{w}, b)$ can be stated as a convex quadratic optimization problem with a unique solution: *minimize* $||\mathbf{w}||$, *subject to the constraints* (one for each training example):

$$y_i(< \mathbf{w}.x_i > +b) \geq 1 \tag{3}$$

The SVM model has an equivalent dual formulation, characterized by a weight vector α and a bias b. In this case, α contains one weight for each training vector, indicating the importance of this vector in the solution. Vectors with non null weights are called *support vectors*. The dual classification rule is:

$$f(\mathbf{x}, \alpha, b) = \sum_{i=1}^{N} y_i \alpha_i < \mathbf{x}_i.\mathbf{x} > +b \tag{4}$$

The α vector can be calculated also as a quadratic optimization problem. Given the optimal α^* vector of the dual quadratic optimization problem, the weight vector $\mathbf{w}^*$ that realizes the maximal margin hyperplane is calculated as:

$$\mathbf{w}^* = \sum_{i=1}^{N} y_i \alpha_i^* \mathbf{x}_i \tag{5}$$

The b^* has also a simple expression in terms of $\mathbf{w}^*$ and the training examples $(\mathbf{x}_i, y_i)_{i=1}^{N}$.

The advantage of the dual formulation is that efficient learning of non-linear SVM separators, by introducing *kernel functions*. Technically, a *kernel function* calculates a dot product between two vectors that have been (non linearly) mapped into a high dimensional feature space. Since there is no need to perform this mapping explicitly, the training is still feasible although the dimension of the real feature space can be very high or even infinite.

By simply substituting every dot product of $\mathbf{x}_i$ and $\mathbf{x}_j$ in dual form with any *kernel function* $K(\mathbf{x}_i, \mathbf{x}_j)$, SVMs can handle non-linear hypotheses. Among the many kinds of *kernel functions* available, we will focus on the d-th *polynomial kernel*:

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i.\mathbf{x}_j + 1)^d$$

Use of d-th polynomial kernel function allows us to build an optimal separating hyperplane which takes into account all combination of features up to d.

We develop our system using SVM [4,7] which perform classification by constructing an N-dimensional hyperplane that optimally separates data into two categories. We have used YamCha[4] toolkit, an SVM based tool for detecting classes in documents and formulating the gene mention detection task as a sequential labeling problem. We use TinySVM-0.07[5] classifier for classification and the *polynomial kernel function*.

2.2 Named Entity Features

Feature selection plays an important role for the success of machine learning techniques. We came up with a variety of features for constructing various models based on the SVM framework. Most of these features are easy to derive and don't require deep domain knowledge and/or external resources for their generation. Due to the use of variety of features, the individual classifiers achieve very high accuracies.

1. **Context words:** These are the words occurring within the context window $w_{i-3}^{i+3} = w_{i-3} \ldots w_{i+3}$, $w_{i-2}^{i+2} = w_{i-2} \ldots w_{i+2}$ and $w_{i-1}^{i+1} = w_{i-1} \ldots w_{i+1}$, where w_i is the current word. This feature is considered with the observation that surrounding words carry effective information for identification of NEs.
2. **Word prefix and suffix:** These are the word prefix and suffix character sequences of length up to n. The sequences are stripped from the leftmost (prefix) and rightmost (suffix) positions of the words. We set the feature values to 'undefined' if either the length of w_i is less than or equal to $n - 1$, w_i is a punctuation symbol or if it contains any special symbol or digit. We experiment with n=3 (i.e., 6 features) and 4 (i.e., 8 features) both.
3. **Word length:** We define a binary valued feature that fires if the length of w_i is greater than a pre-defined threshold. Here, the threshold value is set to 5. This feature captures the fact that short words are likely not to be NEs.
4. **Infrequent word:** A list is compiled from the training data by considering the words that appear less frequently than a predetermined threshold. The threshold value depends on the size of the dataset. Here, we consider the words having less than 10 occurrences in the training data. Now, a feature is defined that fires if w_i occurs in the compiled list. This is based on the observation that more frequently occurring words are rarely the NEs.
5. **Part of Speech (PoS) information:** POS information is a critical feature for NERC. In this work, we use POS information of the current and/or the surrounding token(s) as the features. This information is available with the data.

---

[4] http://chasen-org/ taku/software/yamcha/

[5] http://cl.aist-nara.ac.jp/ taku-ku/software/TinySVM

6. **Dynamic feature:** Dynamic feature denotes the output tags $t_{i-3}t_{i-2}t_{i-1}$, $t_{i-2}t_{i-1}$, t_{i-1} of the word $w_{i-3}w_{i-2}w_{i-1}$, $w_{i-2}w_{i-1}$, w_{i-1} preceding w_i in the sequence w_1^n.
7. **Word normalization:** We define two different types of features for word normalization. The first type of feature attempts to reduce a word to its stem or root form. This helps to handle the words containing plural forms, verb inflections, hyphen, and alphanumeric letters. The second type of feature indicates how a target word is orthographically constructed. Word shapes refer to the mapping of each word to their equivalence classes. Here each capitalized character of the word is replaced by 'A', small characters are replaced by 'a' and all consecutive digits are replaced by '0'. For example, 'IL' is normalized to 'AA', 'IL-2' is normalized to 'AA-0' and 'IL-88' is also normalized to 'AA-0'.
8. **Orthographic features:** We define a number of orthographic features depending upon the contents of the wordforms. Several binary features are defined which use capitalization and digit information. These features are: initial capital, all capital, capital in inner, initial capital then mix, only digit, digit with special character, initial digit then alphabetic, digit in inner. The presence of some special characters like (',','-','.',')','(' etc.) is very much helpful to detect NEs, especially in biomedical domain. For example, many biomedical NEs have '-' (hyphen) in their construction. Some of these special characters are also important to detect boundaries of NEs. We also use the features that check the presence of ATGC sequence and stop words. The complete list of orthographic features is shown in Table 1.

Table 1. Orthographic features

Feature	Example	Feature	Example
InitCap	Src	AllCaps	EBNA, LMP
InCap	mAb	CapMixAlpha	NFkappaB, EpoR
DigitOnly	1, 123	DigitSpecial	12-3
DigitAlpha	2× NFkappaB, 2A	AlphaDigitAlpha	IL23R, EIA
Hyphen	-	CapLowAlpha	Src, Ras, Epo
CapsAndDigits	32Dc13	RomanNumeral	I, II
StopWord	at, in	ATGCSeq	CCGCCC, ATAGAT
AlphaDigit	p50, p65	DigitCommaDigit	1,28
GreekLetter	alpha, beta	LowMixAlpha	mRNA, mAb

3 Datasets, Evaluation Results and Discussions

In this section we present the details of datasets and evaluation results.

3.1 Datasets and Evaluation Technique

We evaluate our proposed approach with the GENETAG training and test datasets, available at the site[6]. GENETAG covers a more general domain of PubMed. It contains

---

[6] ftp://ftp.ncbi.nlm.nih.gov/pub/tanabe/GENEATG.tar.gz

Table 2. Evaluation results on GENETAG datasets (training: GENETAG, test: GENETAG) with various feature subsets. Here, the following abbreviations are used: 'CW':Context words, 'PS': Size of the prefix, 'SS': Size of the suffix, 'WL': Word length, 'IW': Infrequent word, 'NO': Normalization feature, 'PoS': PoS information, 'OR': Orthographic feature, 'Dyn': Dynamic feature, [-i, j]: context words spanning from the left ith position to the jth right position, All[-i,j]: All feature combinations within the context except dynamic NE for the left ith positions, X: Denotes the presence of the corresponding feature, 'r': recall, 'p': precision, 'F': F-measure (we report percentages).

Classi fiers	CW	PS	SS	WL	IW	NO	OR	POS	Dyn	r	p	F
SVM_1	[-1,+1]	4	4	X	X	X	X	X	-1	93.60	91.51	92.54
SVM_2	[-2,+2]	4	4	X	X	X	X	X	-2	94.27	93.21	93.74
SVM_3	[-2,+2]	4	4	X	X	X	X	[-2,+2]	-2	94.20	93.36	93.65
SVM_4	[-2,+2]	4	4	X	X	X	X	[-2,0]	-2	94.41	93.50	93.95
SVM_5	[-2,+2]	3	3	X	X	X	X	X	-2	94.14	93.14	93.64
SVM_6	[-2,+2]	3	3	X	X	X	X	[-2,+2]	-2	94.07	93.32	93.69
SVM_7	[-2,+2]	3	3	X	X	X	X	[-2,0]	-2	94.10	93.29	93.69
SVM_8	[-3,+3]	4	4	X	X	X	X	X	-3	94.17	93.14	93.65

both true and false gene or protein names in a variety of contexts. In GENETAG, not all the sentences of abstracts were included, rather more NE informative sentences were considered. GENETAG selected longer text fragments as entity reference. GENETAG also includes the semantic category word 'protein' for protein annotation. GENETAG is more inclined to select more descriptive expressions for protein annotations. During annotations of GENETAG corpus, some semantic constraints were chosen to make sure that tagged entities must contain their true meanings in the sentence contexts. Based on the gene names from GenBank[7], the GENETAG corpus includes domains, complexes, subunits, and promoters when the annotated entities refer to specific genes/proteins.

Gene mentions in both the training and test datasets were annotated with the 'NEWGENE' tag and the overlapping gene mentions were distinguished by another tag 'NEWGENE1'. However, in this work, we use the standard BIO notations (as in GENIA corpus) to properly denote the boundaries of gene names, and we replace all the 'NEWGENE1' tags by 'NEWGENE' for training and testing. The training dataset contains 7,500 sentences with 8,881 gene mentions. The average length per protein (or, gene) mention is 2.1 tokens. The test dataset consists of 2,500 sentences with 2,986 gene mentions. The system is evaluated using the evaluation script that was provided by the BioCreative-II[8] evaluation challenge for the gene mention detection task.

All the classifiers are evaluated in terms of recall, precision and F-measure. The definitions of recall and precision are given below:

$$\text{recall} = \frac{\text{Number of correctly found NE chunks}}{\text{Number of NE chunks in the gold standard test data}}$$

$$\text{and precision} = \frac{\text{Number of correctly found NE chunks}}{\text{Number of identified NE chunks}}.$$

---

[7] http://www.ncbi.nlm.nih.gov/Genbank/

[8] http://www.biocreative.org/news/biocreative-ii/

The value of the metric F-measure, which is the weighted harmonic mean of recall and precision, is calculated as below:

$$F_\beta = \frac{(1+\beta^2)(\text{recall} + \text{precision})}{\beta^2 \times \text{precision} + \text{recall}}, \quad \beta = 1$$

These are the modified versions of the CoNLL-2003 shared task [6] evaluation script. The script outputs three sets of F-measures according to exact boundary match, right and left boundary matching. In the right boundary matching only right boundaries of entities are considered without matching left boundaries and vice versa. We use the same strict matching criterion that was followed in the Biocreative-II shared task evaluation script[9].

3.2 Results and Discussions

Results of the individual models are reported in Table 2. Each of these models is trained with the subset of features as described in Subsection 2.2. The highest performance corresponds to SVM_4 classifier that yields the overall recall, precision and F-measure values of 94.41%, 93.50%, and 93.95%, respectively. The consistent good performance in each of the models shows the efficacy of the effective feature sets that we identified. In order to obtain higher performance, we combine various SVM based models using the following two voting techniques.

- *Majority Voting* : In this model, all the individual classifiers are combined together into a final system based on the majority voting of the output class labels. If all the outputs differ then anyone is selected randomly.
- *Weighted Voting* : This is a weighted voting approach. In each classifier, weights are calculated based on the average F-measure value of the 3-fold cross validation on the training data.

The majority voted technique achieves the overall recall, precision and F-measure values of 94.61%, 94.10%, and 94.35%, respectively. The weighted vote based technique provides the overall recall, precision and F-measure values of 94.95%, 94.32% and 94.63%, respectively. Results show that the voted approach achieves the performance level, higher than the individual classifiers. To the best of our knowledge this is so far the highest performance in comparison to any previously developed systems that made use of the same datasets.

Table 3. Overall evaluation results on GENETAG datasets (training: GENETAG, test: GENETAG) we report percentages)

Classification Scheme	recall	precision	F-measure
Best individual classifier	94.41	93.50	93.95
Majority Voted Approach	94.61	94.10	94.35
Weighted Voted Approach	94.95	94.32	94.63

---

[9] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/ERtask/report.html

4 Conclusion and Future Works

In this paper we have proposed a voted approach for gene mention detection task. We have used SVM framework as the base classifiers to generate different classification models by varying the available features. We came up with a very rich feature set that itself can achieve very high accuracy. The most important characteristics of our system is that all the features are mostly identified and developed without using any deep domain knowledge and/or external resources. Evaluation results with the benchmark datasets of GENTAG corpus showed the overall recall, precision and F-measure values of 94.41%, 93.50%, and 93.95%, respectively. Final recall, precision and F-measure values after applying the weighted vote based ensemble technique on the SVM based models are 94.95%, 94.32% and 94.63%, respectively. This shows that our approach achieves the state-of-the-art performance. In future we would like to apply some other statistical machine learning algorithms like conditional random field (CRF) and maximum entropy (ME) for solving this gene mention task. Future work also includes use of some new ensemble techniques for combing the outputs of classifiers.

References

1. Aronson, A.R., Bodenreider, O., Chang, H.F., Humphrey, S.M., Mork, J.G., Nelson, S.J., Rindflesch, T.C., Wilbur, W.J.: The NLM Indexing Initiative. In: Proceedings of 2000 AMIA Annual Fall Symposium (2000)
2. Finkel, J., Dingare, S., Manning, C., Nissim, M., Alex, B., Grover, C.: Exploring the boundaries: gene and protein identification in biomedical text. BMC Bioinformatics 6 (2005)
3. Hirschman, L., Yeh, A., Blaschke, C., Valencia, A.: Overview of BioCreAtIvE: critical assessment of information extraction for biology. BMC Bioinformatics 6 (2005)
4. Joachims, T.: Making Large Scale SVM Learning Practical, pp. 169–184. MIT Press, Cambridge (1999)
5. Taira, H., Haruno, M.: Feature Selection in SVM Text Categorization. In: Proceedings of AAAI 1999 (1999)
6. Tjong Kim Sang, E.F., De Meulder, F.: Introduction to the Conll-2003 Shared Task: Language Independent Named Entity Recognition. In: Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003, pp. 142–147 (2003)
7. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer-Verlag New York, Inc. (1995)

Incorporating Fuzzy Trust in Collaborative Filtering Based Recommender Systems

Vibhor Kant and Kamal K. Bharadwaj

School of Computer and Systems Sciences,
Jawaharlal Nehru University, New Delhi, India
{vibhor.kant,kbharadwaj}@gmail.com

Abstract. Collaborative filtering based recommender system (CF-RS) provides personalized recommendations to users utilizing the experiences and opinions of their nearest neighbours. Although, collaborative filtering (CF) is the most successful and widely implemented filtering, data sparsity is still a major concern. In this work, we have proposed a fuzzy trust propagation scheme to alleviate the sparsity problem. Since trust is often a gradual trend, so trust to a person can be expressed more naturally using linguistic expressions. In this work, fuzzy trust is represented by linguistic terms rather than numerical values. We discuss the basic trust concepts such as fuzzy trust modeling, propagation and aggregation operators. An empirical evaluation of the proposed scheme on well known MovieLens dataset shows that fuzzy trust propagation allows reducing the sparsity problem of RSs while preserving the quality of recommendations.

Keywords: Recommender System, Collaborative Filtering, Trust Network, Trust Propagation, Fuzzy Trust.

1 Introduction

The explosive growth of e-commerce and web-based services has made the question of the exploration of information and selection progressively more serious; users are overloaded by options to judge and they may not have the time or knowledge to evaluate these options personally. Recommender systems (RSs) have been proposed to suggest relevant items for online users to cope with the information overload and have become one of the most popular tools in electronic commerce and web-based services [1] [3].

Recommender systems employ mainly three information filtering techniques, collaborative filtering, content-based filtering (CBF) and hybrid filtering techniques [1]. CF is the most widely used technique for web-based RSs. A CF system searches for similar users and then uses ratings from this set of users to predict items that will be liked by the current user [1][9]. Although CF has been used effectively in a wide range of applications, data sparsity is still a major concern.

Recent research on trust aware RSs [9] has shown that they are more robust against sparsity and are more capable of making recommendations. There exist several algorithms for computing, propagating and aggregating the trust in the trust network

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 433–440, 2011.
© Springer-Verlag Berlin Heidelberg 2011

[4][5][6][7][8][10][11][12] etc. Golbeck [5] described a comprehensive review of trust on the Web and proposed algorithms for trust propagation. Josang [7] developed trust propagation operators on the basis of Subjective Logic. Guha et. al., [6] was the first one who utilized the idea of transitivity of distrust and developed a framework for trust propagation [11][12].

Sometimes crisp modeling of trust is not enough for inferring accurate information especially in contradictory situation and also people naturally use linguistic expressions rather than numeric values to describe their trust. Fuzzy sets seem to be an ideal choice for trust modeling. Here, we proposed a fuzzy trust based CF recommender system that suggests items to users in an effective way. The main contribution of our work is fuzzy trust modeling, propagation and aggregation operators. We presented an experimental analysis on MovieLens dataset and show that our work does alleviate the sparsity problem associated with traditional CF.

The rest of this paper is organized as follows: Section 2 provides the fuzzy trust modeling, while section 3 describes the overall framework of our proposed work. Our experimental analysis is presented in section 4 and last section concludes the paper with future work.

2 Fuzzy Trust Modeling

Generally, in trust networks, users specify their trusts to other persons in numerical values. For example, in seven scale ratings, for pessimistic users, a rating of 7 may mean highly trusted but for optimistic users it may mean somewhat high trusted. Is the difference between 2 and 3 the same as the difference 5 and 6? These all contribute to fuzziness that arises from the subjective nature of human. Therefore, trust is treated as a fuzzy variable so there is a need to develop gradual models of trust that quantify the degree to which users may trust each other. We quantified trust into seven triangular fuzzy numbers extremely low trust (ELT), very low trust (VLT), low trust (LT), average trust (AT), high trust (HT), very high trust (VHT) and extremely high trust (EHT) as shown in Fig. 1, with the following membership functions:

$$\mathrm{T_{ELT}(x)} = \left\{\begin{matrix} 2-\mathrm{x}, & \mathrm{x} \le 2 \\ 0, & \mathrm{x} > 2 \end{matrix}\right\} \tag{1}$$

$$\mathrm{T_{a(i)}(x)} = \left\{\begin{matrix} 0, & \mathrm{x} \le \mathrm{i}-2, \mathrm{x} > \mathrm{i} \\ \mathrm{x}-\mathrm{i}+2, & \mathrm{i}-2 < \mathrm{x} \le \mathrm{i}-1 \\ \mathrm{i}-\mathrm{x}, & \mathrm{i}-1 < \mathrm{x} \le \mathrm{i} \end{matrix}\right\} \tag{2}$$

Here $\mathrm{a(i)} = \mathrm{VLT, LT, AT, HT, VHT}$ and EHT for i=3, 4, 5, 6 and 7 respectively.

$$\mathrm{T_{EHT}(x)} = \left\{\begin{matrix} 0, & \mathrm{x} \le 6 \\ \mathrm{x}-6, & 6 < x \le 7 \end{matrix}\right\} \tag{3}$$

Fuzzy trust network is shown in Fig. 2. in which users describe trust ratings to each other employing linguistic terms.

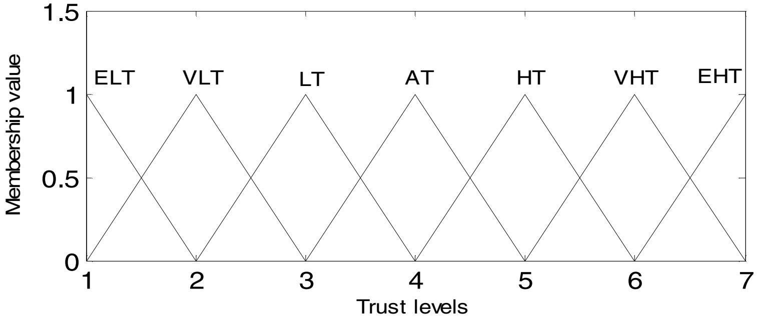


Fig. 1. Membership functions for trust

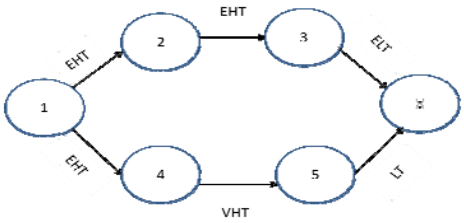


Fig. 2. Fuzzy trust network

3 Incorporation of Fuzzy Trust in CF-RS

In this section, we proposed fuzzy trust propagation and aggregation operators and developed fuzzy trust based recommendations framework.

3.1 Fuzzy Trust Propagation and Aggregation Operators

Propagation operators are used to handle the problem of establishing trust information in unknown user using directly connected trusted users for current user. If a trusts b and b trusts c, this information is used to establish trust in c for a. The only problem in trust propagation is that the propagation trust is not perfectly transitive. For example if a trusts b completely and b trusts c completely, then it has no evidence that a trusts c completely. Hence trust propagation can be considered as a certain degree of fluctuation [10]. Propagation and aggregation operators for fuzzy trust are as follows:

Max-Min Operator

$$S_1 o S_2 = \{[(x,z), \max\{\min\{\mu_{S_1}(x,y), \mu_{S_2}(y,z)\}\}] | x \epsilon X, y \epsilon Y, z \epsilon Z\} \tag{4}$$

Max-Mean Operator

$$S_1 o S_2 = \left\{\left[(x,z), \max\left\{{}^1\!/_2 \times \left\{\mu_{S_1}(x,y) + \mu_{S_2}(y,z)\right\}\right\}\right] \middle| x \epsilon X, y \epsilon Y, z \epsilon Z\right\} \tag{5}$$

where S_1 and S_2 are the fuzzy relations on $X \times Y$ and $Y \times Z$ respectively. If we implement the computation of trust with max-min operator then propagation utilizes the MIN operator and aggregation the MAX operator.

Proposed Operators. In this subsection, a fuzzy trust model is proposed to evaluate trust ratings between two nodes as source and sink. This model consists of two algorithms, one algorithm for propagation of fuzzy trust and another for aggregation. The proposed technique is hereafter referred to as New-Op.

Propagation Operator. This operator is fuzzification of the operator proposed by [10] which is based upon the certain percentage of fluctuation of the rating. In our trust network, the rating of trust from one node to other node is assigned to a linguistic variable in the range of [ELT, EHT]. If a person (node) does not know another, then the operator tries to find out estimated trust rating using a chain of their friends. In each chain, the last person knows the trustee directly. Then the trust rating for trustee depends upon the last rating. But this rating is also influenced by the other intermediary nodes. If these nodes are trustworthy, the source may accept the last rating with high confidence. The main steps involved in this operator for each path in the trust network are as follows:

(a) The suggested rating (R) is the last rating in the trust network.
(b) Compute minimum rating (R_{min}) among other intermediary nodes except last rating using Zadeh's extension principle [13]
(c) Compute heights h_1, h_2 and h_3 of fuzzy sets EHT, ELT and R_{min} respectively[13]
(d) Since rating R may be too exaggerated for optimistic users or too conservative for pessimistic users, therefore compute the amount of fluctuation in R is as follows: For optimistic users, probability of fluctuation in R is 0 and for pessimistic user the fluctuation will be at most absolutely $\frac{h_1 - h_3}{h_1 - h_2}$ then on average, Probability of fluctuation and error in R are computed as

$$\mathrm{pr(Fluct)} = \frac{h_1 - h_3}{2(h_1 - h_2)} \tag{6}$$

$$\mathrm{error} = \frac{\mathrm{EHT} - R_{min}}{2} \tag{7}$$

(e) If $R \geq AT$ then $R' = R - \mathrm{error}$ otherwise $R' = R + \mathrm{error}$ (8)
(f) The estimated trust is given by

$$\mathrm{Trust} = [\mathrm{pr(Fluct)} * R' + \{1 - \mathrm{pr(Fluct)}\} * R] \tag{9}$$

(g) Compute similarity between Trust (9) and each of the linguistic sets defined in section 2. Since all these sets are triangular fuzzy numbers so similarity between two triangular fuzzy numbers a and b is given by

$$\mathrm{sim}(a, b) = 1 - \frac{\{\sum_{i=1}^{3} |a_i - b_i|\}}{3} \tag{10}$$

where $a = (a_1, a_2, a_3)$ & $b = (b_1, b_2, b_3)$

(h) Final trust rating between two nodes is the most similar one to estimated Trust.

Aggregation Operator. If there are several paths between source and sink such as shown in Fig.2, then it is a need to have a method for aggregating. We used the most common operator MAX [13] for aggregating the trusted ratings from different paths.

An example. We show how to compute the final trust rating between node 1 and node 6 shown in Fig 2 using New-Op operator. There are two paths from node 1 to node 6: 1-2-3-6 and 1-4-5-6 For the First path, the suggested rating is R=ELT (1, 1, 2) using step(a) ; R_{min}= EHT (6, 7, 7) using MIN operator [13] in step(b); h_1=7, h_2=1, and h_3=7 be the heights of EHT, ELT and R_{min} ; $\Pr(Fluct)$=0 and error = (0,0,0) using step (d); Now R'= (1,1,2) and *Trust*= (1,1,2) from steps (e) and (f). after it, the trust rating, the most similar one linguistic set to the (1,1,2) is ELT using formula(10). Similarly, for the second path, the trust rating between node 1 and node 6 is LT. Aggregation can be done using MAX operator [13]. Output of this algorithm, the final trust rating between node 1 and node 6 is LT.

3.2 Fuzzy Trust Based Recommendation Framework

The overall system takes as input the trust matrix or the rating matrix and outputs a matrix of predicted ratings, then the RS selects from the row of predictive ratings relative to the user. Firstly, we apply the traditional CF algorithm on MovieLens dataset. In order to make recommendations to the active users, CF performs three steps:

- It compares the active user's ratings with other users in the dataset. It can be done by computing similarity between users using Pearson formula[1]
- It predicts ratings for unseen items to the active user on the basis of the most similar users (neighbours). [1]
- It recommends items to the user with highest predictive ratings

Secondly, we incorporate fuzzy trust in the above CF-RS to deal with sparsity problem. Due to data sparsity, in many cases neighbourhood size is very small and therefore accurate predictions cannot be done. When sparsity level is high, sometimes the neighbourhood set can be just empty and no predictions would be possible. We apply fuzzy trust propagation as discussed above in subsection in 3.1, to enlarge the neighbourhood size that results in better recommendations.

At last, the predicted rating of item *i* for the active user ac is the weighted sum of the ratings given to item *i* by *n* neighbours using Resnick formula [1]

$$p_{ac,i} = \bar{r}_{ac} + \frac{\sum_{u=1}^{n} w_{ac,u}(r_{u,i}-\bar{r}_u)}{\sum_{u=1}^{n}|w_{ac,u}|} \tag{11}$$

Here neighbours can be taken from the user similarity matrix generated by Pearson formula or from the estimated trust matrix generated by propagation operators.

4 Experimental Results

To demonstrate the effectiveness of proposed operators of fuzzy trust, we conducted experiments on the popular MovieLens dataset.

4.1 Design of Experiments

The MovieLens dataset consists of 100,000 ratings (1-5) provided by 943 users on 1682 movies; and each user has rated at least 20 and at most 737 movies. For our experiments we choose three subsets from the data, containing 50,100 and 150 users called ML50, ML100 and ML150 respectively. Since there is no availability of trust statements in MovieLens dataset, we randomly generate the trust matrix consisting linguistic ratings for these datasets. This is to illustrate the effectiveness of the proposed work under varying number of participating users. Each of the datasets was randomly split into 60% training data and 40% test data. The ratings of the items in test data are considered as items unseen by the active user, while the ratings in the training set is used for neighbourhood construction.

In order to test the performance of our scheme, we measure system accuracy using the mean absolute error (MAE) and coverage of the system. MAE measures the average absolute deviation of predicting rating of an item from the actual rating for the item. Coverage is measured as number of items for which RSs can generate prediction over total number of unseen items.

4.2 Results

To demonstrate the ability of the proposed scheme New-Op-CF to offer better accuracy we compare the MAE and coverage with CF, Max-Min-CF, and Max-Mean-CF. The results are presented in Table 1. The MAE is computed based on the average over 20 runs of the experiments over different datasets. A lower value of MAE corresponds to a better performance. It is clear from table 1 the proposed scheme New-Op-CF considerably outperforms other methods.

Table 1. MAE comparison of New-Op-CF with CF, Max-Min-CF and Max- Mean-CF

Datasets	CF	Max-Min-CF	Max-Mean-CF	New-Op-CF
ML50 MAE	0.8960	0.8735	0.8735	0.8744
ML100 MAE	0.9194	0.9066	0.9065	0.8930
ML150 MAE	0.8853	0.8888	0.8887	0.8668

The total coverage for New-Op-CF is always greater than the other techniques shown in Table 2. A higher value of coverage implies the better performance of the proposed scheme. The MAE for the different runs of the experiment for ML50 is shown in Fig. 3. A total of 20 runs were made for all datasets. The proposed method New-Op-CF performed better than any of the other techniques in the terms of predictive accuracy as well as coverage.

Table 2. Comparison of Coverage of New-Op-CF with CF, Max-Min-CF and Max-Mean-CF

Datasets	CF	Max-Min-CF	Max-Mean-CF	New-Op-CF
ML50 Coverage	86.7	87.2	87.4	88.6
ML100 Coverage	87.7	87.9	87.9	88.5
ML150 Coverage	88.8	89.2	90.2	90.5

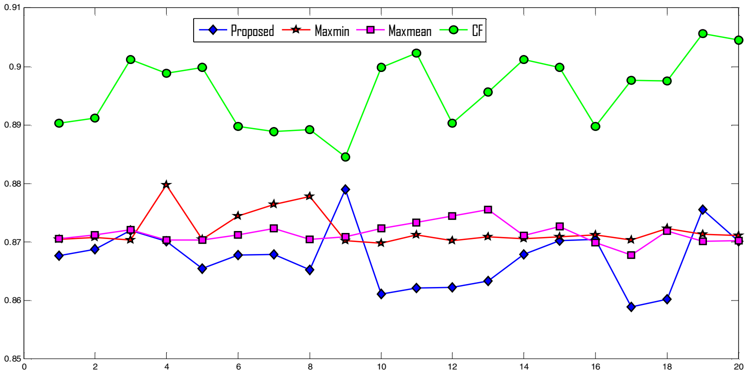


Fig. 3. MAE for ML50 over 20 runs

5 Conclusions and Future Work

This research proposes modeling of fuzzy trust and corresponding propagation operators to deal with the sparsity problem. Incorporation of fuzzy trust network among users has resulted in alleviating data sparsity problem, thereby producing quality recommendations. Through experimental results, we have clearly demonstrated that New-Op-CF outperforms traditional CF, Max-Min-CF and Max Mean-CF.

Our future works goes in several directions. First, we will focus on developing different efficient techniques to compute and propagate fuzzy trust [11][12]. Second, it is to be seen how two level filtering based on both trust and reputation [4] and utilization of various sparsity measures [2] can be incorporated in the fuzzy trust enhanced CF system to further improve its performance.

References

1. Adomavicius, G., Tuzhilin, A.: Toward The Next Generation of Recommender Systems: A Survey of The State-of-The-Art and Possible Extensions. IEEE Trans. Knowledge and Data Engineering 17(6), 734–749 (2005)
2. Anand, D., Bharadwaj, K.K.: Utilizing Various Sparsity Measures for Enhancing Accuracy of Collaborative Recommender Systems Based on Local and Global Similarities. Expert Systems with Applications 38(5), 5101–5109 (2010)

3. Al-Shamri, M.Y.H., Bharadwaj, K.K.: Fuzzy-Genetic Approach to Recommender System Based on A Novel Hybrid User Model. Expert Systems with Applications 35(3), 1386–1399 (2008)
4. Bharadwaj, K.K., Al-Shamri, M.Y.H.: Fuzzy Computational Models for Trust and Reputation Systems. Electronic Commerce Research and Applications 8(1), 37–47 (2009)
5. Golbeck, J.: Trust and Nuanced Profile Similarity in Online Social Networks. ACM Transactions on the Web (TWEB) 3(4), 1–33 (2009)
6. Guha, R., Kumar, R., Raghavan, P., Tomkins, A.: Propagation of Trust and Distrust. In: Proceedings of the 13th International Conference on World Wide Web, New York, pp. 403–412 (2004)
7. Jøsang, A., Marsh, S., Pope, S.: Exploring Different Types of Trust Propagation. In: Stølen, K., Winsborough, W.H., Martinelli, F., Massacci, F. (eds.) iTrust 2006. LNCS, vol. 3986, pp. 179–192. Springer, Heidelberg (2006)
8. Lesani, M., Montazeri, N.: Fuzzy Trust Aggregation and Personalized Trust Inference in Virtual Social Networks. Computational Intelligence 25(2), 51–83 (2009)
9. Massa, P., Avesani, P.: Trust-aware Collaborative Filtering for Recommender Systems. In: Meersman, R. (ed.) OTM 2004. LNCS, vol. 3290, pp. 492–508. Springer, Heidelberg (2004)
10. Shekarpour, S., Katebi, S.D.: Modeling and Evaluation of Trust with an Extension in Semantic Web. Web Semantics: Science, Services and Agents on the World Wide Web 8, 26–36 (2010)
11. Victor, P., Cornelis, C., De Cock, M., Da Silva, P.P.: Gradual Trust and Distrust in Recommender systems. Fuzzy Sets and Systems 160(10), 1367–1382 (2008)
12. Victor, P., Cornelis, C., De Cock, M., Da Silva, P.P.: Practical Aggregation Operators for Gradual Trust and Distrust. Fuzzy Sets and Systems (article in press, corrected proof, 2011)
13. Zadeh, L.A.: Fuzzy Sets. Information Control 8, 338–353 (1965)

A Function Based Fuzzy Controller for VSC-HVDC System to Enhance Transient Stability of AC/DC Power System

Niranjan Nayak, Sangram Kesari Routray, and Pravat Kumar Rout

Multidisciplinary Research Centre, S.O.A University Bhubaneswar, India
`Pkrout_india@yahoo.com,`
`{routraysk,niranjannayak.el.kec}@gmail.com,`

Abstract. This paper presents a robust function based optimized fuzzy controller for VSC-HVDC transmission link operating in parallel with an AC transmission line connecting a synchronous generator to infinite bus. A two input one output methodology based optimized non-linear fuzzy logic controller has been proposed to provide voltage support by means of reactive control at both ends; to damp power oscillations and improve transient stability by controlling either active or reactive power; and to control the power flow through the HVDC link. An improved Teaching Learning Algorithm is applied to tune the gains of the fuzzy controller. Comprehensive computer simulations are carried out to verify the proposed control scheme under several system disturbances like changes in short-circuit ratio, converter parametric changes, torque change and faults on the converter and inverter buses. Based upon the time domain simulations in MATLAB/SIMULINK environment, the proposed controller is tested and its better performance is shown compare with the conventional PI controllers with respect to voltage stability, damping of power oscillations and improving of transient stability.

1 Introduction

It becomes a challenge to increase power delivery by improving transfer capability with AC expansion options in meshed, heavily loaded ac networks satisfying the technical requirements such as dynamic performances. Fast control of active and reactive power of VSC-HVDC system can improve power grid dynamic performance under disturbances. In case of serious disturbance threatens system transient stability, fast power run-back and even instant power reversal control functions can be used to help maintain synchronized power grid operation[1-3]. VSC-HVDC system can also provide effective damping to mitigate electromechanical oscillations by active and reactive power modulation. A range of advanced control functions can be implemented in VSC-HVDC systems for enhancement of ac network steady-state and dynamic performance. An attempt has been made in this paper in the above mentioned prospective.

VSC-HVDC can be operated in three modes: i) Constant DC voltage control mode ii) Constant active power control mode iii) Constant AC voltage control mode[4]. The

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 441–451, 2011.
© Springer-Verlag Berlin Heidelberg 2011

VSC-HVDC transmission link has four control inputs, namely modulation signal magnitude and phase angle at both the converter stations and their interaction makes system a truly non-linear multi input and multi output control system. The control of all the possibilities is normally achieved through proportional-plus-integral (PI) controllers. These PI controllers suffer from the inadequacies of providing suitable control and transient stability enhancement over a wide range of system operating conditions under various faulted situation.

Several authors have presented mathematical models and control strategies for VSC-HVDC transmission that include small signal stability [5], decoupling control in converter stations using feedback linearization [6], LMI based robust control [7], and adaptive control [8-10]. Most of these papers have considered a point to point VSC-HVDC system without considering a parallel AC line in which case the sending and receiving ends are completely decoupled. Although some research has been conducted for parallel AC-VSC-HVDC systems [11-14], to explore the possibility of their smooth operation, till a lot of aspects are yet to be examined thoroughly. However also these previous studies were at the expense of neglecting a number of very important features in HVDC system operation, particularly related to the HVDC-HVAC interactions.

In this study, a robust nonlinear control is applied to the HVDC Light transmission systems in order to improve the dynamic behavior performance under wide range of operating and faulted conditions. The controller design is based on a simple function based optimized Fuzzy Controller. Here the gains of the Fuzzy Controller are optimized using a modified Teaching Learning Technique. This paper clearly describes the advantages of function based fuzzy controller over conventional PI Controller. The effectiveness of the robust controllers is demonstrated using the simulation studies with the aid of the MATLAB software package. The simulation results show that the controllers contribute significantly to improve the dynamic behavior of the HVDC light transmission system under a wide range of operating conditions, enhance the system stability and damp the undesired oscillations.

The rest of the paper is organized in six sections as follows. In section 2, the time domain mathematical modeling of the VSC-HVDC light transmission system is presented. In section 3, the conventional PI controller and two rules based Fuzzy Logic Controller is briefly explained. The improved Teaching Learning Algorithm is explained in detail in Section 4. Simulation study that illustrates the effectiveness of the proposed strategies is discussed in section 5. At last, conclusions are drawn in Section 6.

2 Mathematical Modeling in D-Q Frame of Reference

In this paper mathematical model of the whole system has been derived in d-q frame. For the simplicity of analysis the three phases of the AC system are assumed to be symmetrical and the voltage of the AC source is in the q-axis exactly. Thus the reactive power can be controlled by the d-axis component and the active power or V_{dc} can be controlled by q axis component.

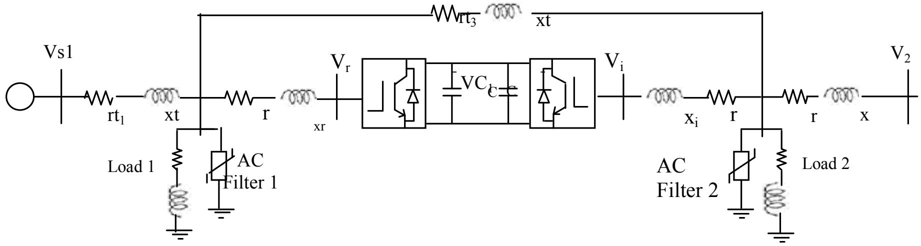


Fig. 1.

The whole model is based upon the state space control theory ($\dot{x} = A.x + B.u$). The controllers have been designed according to the system state. The model studied in this paper is shown above in Fig.1.

The VSC HVDC system shown in fig.1 has a synchronous generator equipped with a voltage stabilizer and a parallel AC-HVDC Light system. x_{t1}, x_{t2}, x_{t3} and x_{t4} are the transformer reactance. x_1 is the line reactance and x_r, x_i are the reactance of the rectifier and inverter respectively. P_{L1}, Q_{L1} And P_{L2}, Q_{L2} are the loads in the passive network. DC line resistance is taken as the R_{dc}. C_1 *and* C_2 Are the DC link capacitors. V_1 And V_R are the node bus and rectifier bus voltage respectively on the converter side network. V_I and V_2 are the inverter bus and infinite bus voltage respectively on the inverter side network.

2.1 Synchronous Generator

The synchronous generator is described by a third order nonlinear mathematical model given by Ref. [15-16]

$$\dot{\delta} = \omega_1 - \omega_0 \qquad \dot{\omega} = \frac{Pm - Pe}{IC} \tag{1}$$

$$\dot{E}'_q = \frac{1}{T'_0}\left\{E'_{q0} + E'_f - E'_d - (x'_d - x'_{dd}).I_{1d}\right\} \tag{2}$$

$$\dot{E}'_f = \frac{1}{T'_e}\left\{- E'_f + k_e\left(V_{1ref} - V\right)\right\} \tag{3}$$

Where δ , ω , E'_f , E'_q are the power angle, angular speed, field voltage and quadrature axis voltage of the generator respectively. The excitation system of the generator consists of a simple AVR along with the supplementary PSS.

2.2 Rectifire Station

The dynamic equation of converter station are given by

$$\dot{I}_{rD} = \frac{V_{1D} - V_{rD}}{L_r} + \omega . I_{rQ} + \frac{-R_r}{L_r} I_{rD} \qquad \dot{I}_{rQ} = \frac{V_{1Q} - V_{rQ}}{L_r} - \omega . I_{rD} + \frac{-R_i}{L_i} I_{iQ} \tag{4}$$

$$\dot{V}_{dc1} = \frac{1}{C}\left\{ \frac{P_{dc1}}{V_{dc1}} - (\frac{V_{dc1} - V_{dc2}}{r_{dc}}) \right\} = \frac{V_{1D} I_{rD} + V_{1Q} I_{rQ}}{C_{dc} V_{dc}} - \frac{I_{dc}}{C_{dc}} \tag{5}$$

WhereV_{1D}, V_{1Q} andV_{rD}, V_{rQ} are the direct and quadrature axis voltage of the ac bus and the rectifier bus respectively. I_{rD}, I_{rQ} are the d-q axes current flowing into the rectifier. I_{dc} is the dc current flowing out of the rectifier. L_r and C_{dc} are the equivalent inductance offered by the rectifier and dc link capacitor respectively.

The modulus index m_1 and firing angle α_1 can be related as

$$V_{rD} = m_1 . V_{dc1} . \sin \alpha_1 \qquad V_{rQ} = m_1 . V_{dc1} . \cos \alpha_1 \tag{6}$$

The active and the reactive powers at the converter and the inverter sides are given by

$$P_r = V_{rd} I_{rd} + V_{rq} I_{rq} \quad Q_r = -V_{rd} I_{rq} + V_{rq} I_{rd} \tag{7}$$

2.3 Inverter Station

The dynamic equations of the inverter are as follows $\dot{I}_{iD} = \frac{V_{iD} - V_{2D}}{L_i} + \omega . I_{iQ}$, $\dot{I}_{iQ} = \frac{V_{iQ} - V_{2Q}}{Li} - \omega . I_{iD}$ Where I_{iD}, I_{iQ} are the d-q axes current flowing out of the inverter? L_i is the equivalent inductance offered by the inverter. The modulus index m_2 and firing angle α_2 can be related as

$$V_{iD} = m_2 . V_{dc2} . \sin \alpha_2 \qquad V_{iQ} = m_2 . V_{dc2} . \cos \alpha_2 \tag{8}$$

The active and the reactive powers at the converter and the inverter sides are given by

$$P_i = V_{id} I_{id} + V_{iq} I_{iq} \qquad Q_i = -V_{id} I_{iq} + V_{iq} I_{id} \tag{9}$$

3 Control Scheme for VSC-HVDC System

Since a parallel AC line is connected to a VSC-HVDC system both the rectifier and inverter stations are coupled giving rise to a system instability for certain operating conditions. Also due to coupling both of the sides are no more independent that is fault on one side reflects to the other. Normally PI controller is used for this purpose for providing damping to the system oscillations. Thus for improving damping performance significantly a novel two rule based fuzzy controller has been proposed that gives better stability to the system under abnormal conditions.

3.1 PI Controller

The PI controller is designed to provide decoupled and independent control of the DC link voltage and the generator side reactive power for converter station and the control of active power and reactive power for inverter station. The state variables of rectifier station are taken as I_{rd}, I_{rq} and V_{dc1}. For convenience dc voltage and reactive power of the rectifier are taken as the output variables. Since the reference bus d-axis voltage is taken as zero, the reactive power directly controlled by I_{rq}. The state variables of inverter station are I_{id}, I_{iq}. Since the inverter station is operating in the active and reactive control mode the output states are taken as I_{id}, I_{iq}. The whole PI controller strategies for rectifier station and inverter station are shown in the Fig.2 and Fig. 3 respectively.

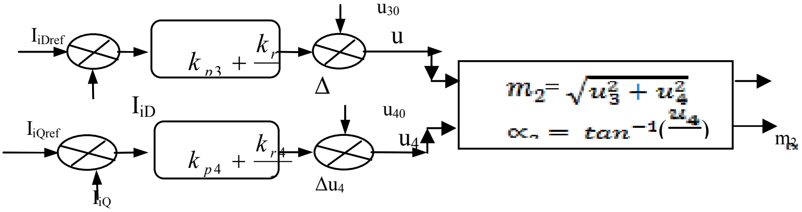

I_{iDref}
$k_{p3} + \frac{k_r}{}$
u30
u
Δ
I_{iD}
I_{iQref}
$k_{p4} + \frac{k_{r4}}{}$
u40
u4
Δu4
I_{iQ}
$m_2 = \sqrt{u_3^2 + u_4^2}$
$\alpha_2 = \tan^{-1}(\frac{u_4}{})$
m_2

Fig. 2. PI controller for the rectifier station

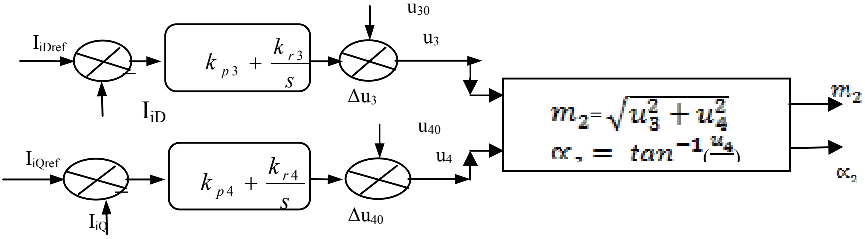

I_{iDref}
$k_{p3} + \frac{k_{r3}}{s}$
u30
u3
Δu3
I_{iD}
I_{iQref}
$k_{p4} + \frac{k_{r4}}{s}$
u40
u4
Δu40
I_{iQ}
$m_2 = \sqrt{u_3^2 + u_4^2}$
$\alpha_2 = \tan^{-1}(\frac{u_4}{})$
m_2
α_2

Fig. 3. PI controller for the inverter station

Here the trial and error technique is avoided and TLBO technique is used to optimize the values of the gains $k_{p1}, k_{r1}, k_{p2}, k_{r2}, k_{p3}, k_{r3}$ *and* k_{p4} . Both of the result has been demonstrated by simulation.

3.2 Two-Rule Based PI Like Fuzzy Controller

To overcome the tuning problem of conventional PI controller, here a two rule based PI like fuzzy controller has been proposed as shown in fig.4. This section describes the main features of the proposed two rule base two input one output FLC.

The proposed FLC uses two inputs as error e, and change of error de, and one output du (change in control). PI like FLC is based upon two rules and one metarule as follows:

R1: If e is EP and de is CEP then du is UP.,R2: If e is EN and de is CEN then du is UN.

In the above rule base, the fuzzy sets for error e, change of error de, and change of control du are EP(error positive), and EN(error negative), CEP(change of error positive), and CEN(change in error negative), UP(control positive),and UN(control negative), respectively. Since the above rule base is not complete as at least four rules are required for the fuzzy PI or PD controller. Thus, a third metarule of the form given below is used to obtain the fuzzy 2-rule based controller:

"*If e is not self correcting, then control action du is not zero and depends upon the sign and magnitude of e and de*". This is achieved by defining the relationship as sign (du) = sign (e). Thus the metarule takes the form "*If sign (e) = sign (de) ≠ 0, Then sign (du) = sign (e)*".

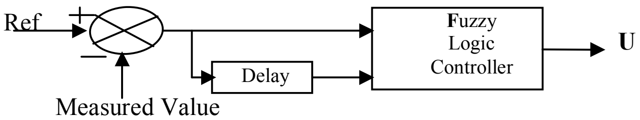


Fig. 4. Generalized Fuzzy like PI controller

The membership functions for the fuzzy sets EP and EN are chosen as $\mu_1 = \mu_{EP}(e) = .5(1+\tan(h(K_1 e))$, ($\mu_{EN}(e) = 1-\mu_1 = .5(1-\tan(h(K_1 e))$ (10)

Similar expressions hold for CEP, and CEN, respectively as $\mu_2 = \mu_{CEP}(de) = .5(1+\tan(h(K_2 de))$, $\mu_{CEN}(de) = 1-\mu_2 = .5(1-\tan(h(K_2 de))$ (11)

The defuzzified output becomes

du = (2/3) G ($(A_1 - A_2)/(A_1 + A_2)$, where $A_1 = 0.5\lambda\mu_1 + 0.5(1-\lambda)\mu_2$,

$$A_2 = 0.5\lambda(1-\mu_1) + 0.5(1-\lambda)(1-\mu_2) \quad (12)$$

Using the above relations and after some simplifications, the defuzzified control output is obtained as du = (2/3) G ($(\lambda \tanh(K_1 e) + (1-\lambda)\tanh(K_2 de))$ (13)

A slightly different version is obtained by using a product-t norm for the defuzzification process in which the value of A_1 becomes equal to $A_1 = A_2$ exp ($K_1 e + K_2 de$), and using simplification du is obtained as

$$du = (2/3)\, G \tanh(K_1 e + K_2 de) \quad (14)$$

The various gains used in the above equations like K_1, K_2, G are to be chosen appropriately for optimal control performance. Thus the control input at time t is

$$u(t) = u(t-1) + \frac{2}{3} G \tanh(K_1 e(t) + K_2 de(t)) \tag{15}$$

Where K_1 acts as integral constant and K_2 as proportional constant. Keeping K_1=0, G and K_2 are tuned. Then K_1 is gradually increased to achieve the required output. In this way all the gains are tuned at one operating condition but for different operating conditions the gains are optimized using the improved teaching learning algorithm. The converter and inverter station auxiliary fuzzy controllers can be written by using equation (30) separately as follows. For the converter station

$$u_{f1}(t) = u_{1f1}(t-1) + \frac{2}{3} G_1 \tanh(K_{11} e_1(t) + K_{12}\ de_1(t)) \text{ , } u_{f2}(t) = u_{f2}(t-1) + \frac{2}{3} G_2 \tanh(K_{21} e_2(t) + K_{22}\ de_2(t)) \tag{16}$$

Where $e_1 = I_{rD}^{ref} - I_{rD}$, $de_1 = e_1(t) - e_1(t-1)$

$$e_2 = V_{dc1}^{ref} - V_{dc1}, de_2 = e_2(t) - e_2(t-1) \tag{17}$$

For inverter station

$$\begin{gathered} u_{f3}(t) = u_{f33}(t-1) + \frac{2}{3} G_3 \tanh(K_{31} e_3(t) + K_{32}\ de_3(t)) \\ u_{f4}(t) = u_{f4}(t-1) + \frac{2}{3} G_4 \tanh(K_{41} e_4(t) + K_{42}\ de_4(t)) \end{gathered} \tag{18}$$

Where $e_3 = I_{iD}^{ref} - I_{iD}$, $de_3 = e_3(t) - e_3(t-1)$ (19)

And $e_4 = I_{iQ}^{ref} - I_{iQ}, de_4 = e_4(t) - e_4(t-1)$ (20)

4 Teaching-Learning Based Optimization (TLBO)

This is a population based method based on the effect of influence of a teacher on learners. Due to faster convergence rate, better searching capability to find global optimum, less adjustment of algorithm parameter change and less computational effort than the other nature inspired algorithm it is used to tune the fuzzy controller parameters.

Teacher Phase: In this phase learning of the learner from the teacher is encapsulated by adding a factor to the old value as follows. The updated values $X_{new,s}$ in the iteration are calculated by adding some percentage of the difference between the teacher and the individual learner subject value to the old value $X_{old,s}$ as follows.

$$X_{new,s} = X_{old,s} + r\ and(X_{teacher,s} - T_f(Mean_{ind\text{-}sub,s})) \tag{21}$$

Learning Phase: In this phase the second way of learning of the learner by interaction with other learners is taken care of by adding a factor to the above value $X_{new,s}$ .as follows. But with a condition a learner learns new if the other learner has more knowledge comparatively.

$$X_{new-final,s} = X_{new,s} + \left(X_i - X_j\right) \text{ if } X_i > X_j \tag{22}$$

$$X_{new-final,s} = X_{new,s} + \left(X_j - X_i\right) \text{ if } X_i < X_j \tag{23}$$

Stopping Criterion: The process terminate once the maximum generation number is achieved. To validate the established steady state model and the proposed control strategy, the operating points are taken into consideration are mentioned in appendix-A for theoretical simulation studies in MATLAB.

The fitness function chosen here for finding optimized PI and Fuzzy controller gains is chosen as

$$F = \frac{1}{1 + \frac{1}{m}\sum_{i=1}^{m}[e_1^2(i) + e_2^2(i) + e_3^2(i) + e_4^2(i)]}$$

5 Simulation Result

Case-1. The system is simulated with a LLLG fault at bus-1 from 1 sec to 1.2 second. Due to the fault the ac voltage at the corresponding bus is decreased to a critical level. The performance of the optimized fuzzy controller restores the system faster than the conventional PI controller. The performance improvement with the new controller shown in Fig.5 clearly indicates the robustness of the proposed controller. The damping property marginally improves along with the reduction in overshoot and settling time.

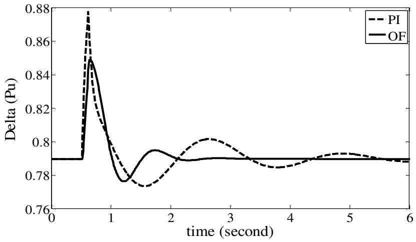


Fig. 5. (a) Variation of power angle

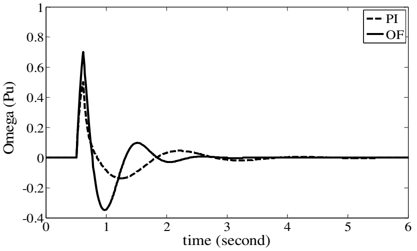


Fig. 5. (b) Variation of rotor speed

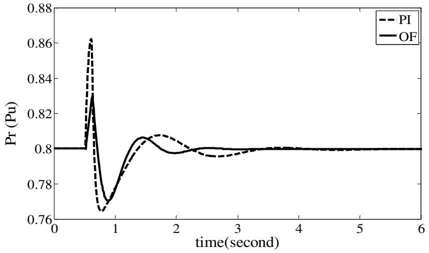


Fig. 5. (c) Variation of rectifier power

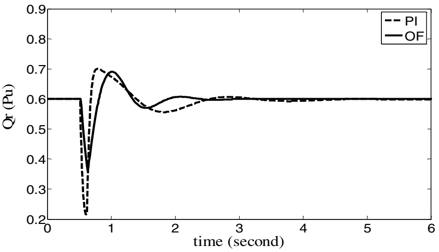


Fig. 5. (d) Variation of Qr at rectifier

Case-2. With operating condition of Pr=0.8Pu andQr=0.6Pu,the 10-cycle LLLG fault is simulated on the inverter side. Fig.6 shows the satisfactory performance of the proposed optimized Fuzzy controller in comparison to the optimized PI controller. The proposed controller maintains a complete coupling effect between the stations. The ac side of the rectifier station operates completely in a stable condition and is very less affected by the fault on the inverter side. Fig. 6 indicates

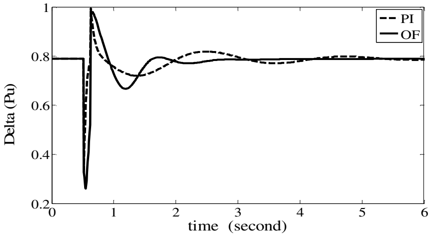


Fig. 6. (a) Variation of power angle

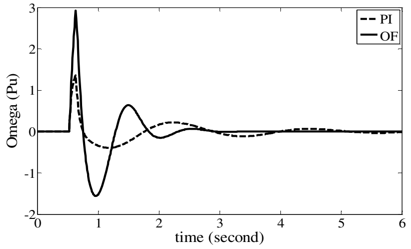


Fig. 6. (b) Variation of rotor speed

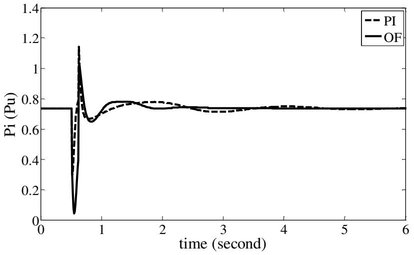


Fig. 6. (c) Variation of Pi at inverter

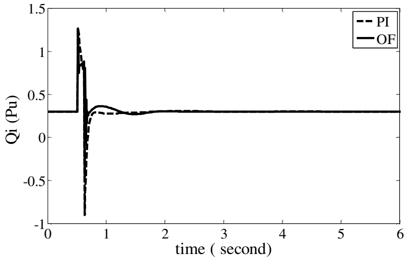


Fig. 6. (d) Variation of Qi after the rectifier

Case-3. Here a 1 cycle fault at bus 1 is created by increasing the mechanical input to the generator (20 percent of the preliminary value) with an operating point Pr=1.03Pu, Qr=0.3pu, is simulated on the inverter side. Fig.7 shows the satisfactory improvement of the power system by the proposed controller.

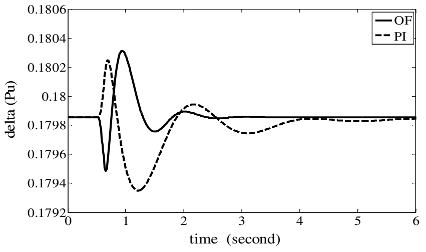


Fig. 7. (a) Variation of power angle

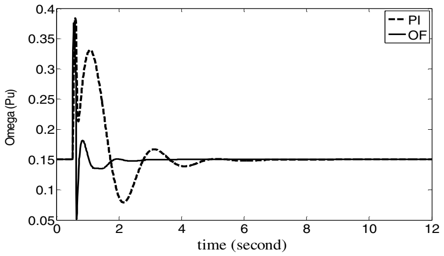


Fig. 7. (b) Variation of rotor speed

6 Conclusion

The VSC based HVDC light system with optimized function based fuzzy controller has been proposed and designed in this paper. The gains of the proposed controller have been tuned using a adaptive Teaching-Learning based optimization technique. The methodology used to design the PI like FLC for the HVDC light system works effectively. Under normal and fault condition the controller is tested which gives the satisfactory result. Simulation result shows that with proposed control strategy, quick response and dynamic stability have been achieved for any kind of changes and high level control accuracy is attained at different operating conditions of the system as well as for a variety of disturbances occurring in the network.

References

[1] Andersen, B.R., Xu, L., Horton, P.J., Cartwright, P.: Topologies for VSC Transmission. Power Engg. Journal 16, 142–150 (2002)

[2] Asplund, G.: Application of HVDC Light to power system enhancement. IEEE Power Eng. Soc. Winter Mtg. 4, 2498–2503 (2000)

[3] Zhang, G., Xu, Z.: Steady-state model for VSC based HVDC system and its controller design. IEEE Power Engineering Society Winter Meeting 3, 1085–1090 (2001)

[4] Durrant, M., Werner, H., Abbott, K.: Synthesis of multi-objective controllers for a VSC HVDC terminal using LMIs. In: IEEE Conference on Decision and Control, pp. 4473–4478 (December 2004)

[5] Ruan, S.Y., Li, G.J., Jiao, X.H., Sun, Y.Z., Lie, T.T.: Adaptive control design for VSC-HVDC system based on backstepping approach. Electric Power System Research 77(5-6), 559–565 (2007)

[6] Ruan, S.X., Li, G.J., Peng, L., Sum, Y.Z., Lie, T.T.: A non-linear control for enhancing HVDC Light transmission system stability. Electrical Power Energy Systems 29, 565–570 (2007)

[7] Xu, L., Anderson, B.R., Cartorite, P.: VSC transmission operating under unbalanced AC conditions- analysis and control design. IEEE Transactions Power Delivery 20(1), 427–434 (2005)

[8] de la Villa Jean, A., Acha, E., Exposito, A.G.: Voltage source converter modeling for power system state estimation: STATCOM and VSC-HVDC. IEEE Transactions Power System 23(4), 1552–1559 (2008)

[9] Latorre, H.F., Ghandhari, M., Soder, L.: Control of a VSC-HVDC Operating in Parallel with AC Transmission Line. In: IEEE Conference (2006)

[10] Li, C., Li, S., Han, F., Shang, J., Qiao, E.: A VSC-HVDC fuzzy controller to damp oscillation of AC/DC Power System. In: IEEE Conference, ICSET 2008, pp. 816–819 (2008)

[11] Latorre, H.F., Ghandhari, M., Soder, L.: Active and Reactive Power Control of VSC-HVDC. Electrical Power System Research 78, 1756–1763 (2008)

[12] Scholtz, E.: Towards realization of a highly controllable transmission system-HVDC light. ABB Corporate Research, 1–21 (2008)

Appendix

Synchronous machine data Generator rating = 900 MVA,Base MVA = 500

P_e = 1.4 pu, Q_e =0.8 pu,,P_t = 0.5 pu,Q_t = 0.2 pu,Pdc =0.3 pu

Load active power P_L =0.7 puLoad reactive power, Q_L =0.3 Pu

$$k_e = 30,\ x_d = 1.8,\ x'_d = 0.3,\ x_q = 1.7,\ \mathrm{H} = 6.5,\ \tau'_{d0}=8$$

PSS data

$$U_{pss} = K_\delta \cdot \frac{ST_q}{1+ST_q} \cdot \left(\frac{1+ST_1}{1+ST_2}\right),\quad K_\delta = 0.24,\ T_q = 0.4,\ \mathrm{T}_1 = 0.03,\ T_2 = 0.01$$

Optimized PI gains k_{p1} = 1.98; k_{r1} = 3.56; k_{p2} = 0.45; k_{r2} = -1.3; k_{p3} = .175; k_{r3} = .656; k_{p4} = 0.39; k_{r4} = -1.57;

Optimized fuzzy control gains K_{11} = 0.341; K_{12} = 4.593; G_1 = 0.37; K_{21} = 0.138; K_{22} = 2.49; G_2 = 0.13; K_{31} = 0.412; K_{32} = - .593; G_3 = 0.63; K_{41} =0.237; K_{42} = 3.92; G_4 = 0.19;

Control parameters of TLBO

Number of particles = 20; Number of variables for optimized fuzzy controller = 12; The maximum and minimum limits for the gains: 10 and -5.

A Bayesian Network Riverine Model Study

Steven Spansel, Louise Perkins, Sumanth Yenduri, and David Holt

University of Southern Mississippi,
730 East Beach Boulevard, Long Beach, MS 39560

Abstract. In this paper we apply a Bayesian Network Methodology to a riverine model study. We show that continuous variables, following Ebert-Uphoff's method, would improve this model, and we modify an existing share-ware Bayesian Network to simulate continuous variables. We also present a new metric for the Bayesian Network.

1 Introduction

Bayesian Networks are software tools that encode causal relationships. Graphically they are represented as Directed Acyclic Graphs (DAG's). Each DAG node represents a random variable that can take on two or more discrete values. Each DAG arc represents a causal conditional dependent relationship between the two nodes; a non-zero covariance relationship is represented only if no other arc implies the same dependence.

Mathematically, two random variables, A and B, are conditionally independent, given C, if any of the conditional probability equations below hold:

$$P(A,B|C) = (A|C)P(B|C) \tag{1}$$

$$P(A|B,C) = P(A|C) \tag{2}$$

$$P(B|A,C) = P(B|C) \tag{3}$$

If we consider the set of discrete values that our stochastic variables can assume as bins, the set of bins must of necessity cover the full range of the continuous variable being discretized. In general these bins form a partition over the values within the range; this is a hard discretization. Bayesian Networks propagate new probability tables for nodes via Bayesian inference, which is based on Bayes' Theorem. For a case where a node representing variable A has n bins, the general form of Bayes' Theorem can be used to find the probability of some A_i, given new evidence E [1]. A third, "chained" version of the theorem can be used to find the probability that some event C has occurred, given that events A and B have occurred [2].

In a multi-connected network $O(n)$ calculations would need to be performed, each themselves requiring linear time, bringing the total computation complexity cost to quadratic time complexity [3]. To speed up this process, Bayesian Network software packages use join tree algorithms [4] to decompose the graph represented by the Bayesian Network into a join tree. After this, a propagation algorithm is executed on the join

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 452–459, 2011.
© Springer-Verlag Berlin Heidelberg 2011

tree. There are many established algorithms [5] for performing this propagation. Many general use, pre-built implementations are available from a variety of sources. For example, there are software development firms that produce high quality commercial products. In addition, several University Computer Science / Mathematics Departments maintain free (with license limitations), open-source implementations.

There are two primary characteristics that separate these implementations. The first is the method of use. Many implementations are only available for certain Operating Systems (most commonly, Microsoft Windows). Some implementations can only be used via a graphical user interface (GUI). Such implementations cannot be automated by incorporating them into a computer program. Still other implementations are not wide enough in their scope. For instance, there are some that focus exclusively on using Bayesian Networks to perform sensitivity analysis. The second distinguishing characteristic among Bayesian Network software packages is their implementation of the junction tree algorithm. In order to be deemed correct, junction tree algorithms must arrive at the correct answer for all computations. However, some implementations of this algorithm perform these calculations much more efficiently than other implementations. This is one of the main areas of competition between competing commercial Bayesian Network software implementations. For a list of some available packages, see Appendix 2 of Agena Risk's Knowledge Transfer Report on Bayesian Network technology [6]. We utilize the Netica Bayesian Network software package, produced by [7]. It is available for all major platforms (Microsoft Windows, Macintosh OS X, Linux, and Sun Solaris). Also, it can be utilized both via a GUI and via Application Programming Interfaces (APIs) available for many different programming languages, including Java, C, C++, .net, and others. Furthermore, it has earned a reputation as being a fast and stable implementation through its performance in other scientific research projects.

We use Netica in two forms: as a Java API and as a Windows GUI. Versions of this software can be downloaded from http://www.norsys.com/netica-j.html and http://www.norsys.com/download.html, respectively. Once downloaded, no installation of the software is necessary. The versions that can be freely downloaded but are limited in size – the number of nodes that can be contained by a network is limited. Furthermore, there are legal restrictions on the use of the software. These can be viewed online at http://www.norsys.com/netica-j.html.

2 Related Research

[8] propose a probability based soft discretization. We present a solution that mimics a soft discretization and works with existing Software Tools. Hard discretization of a variable loses accuracy. Precisely, if a value of x is allocated to a bin with range [x-.5] – [x+.2], the effect of x changes to that of x-1.5 – the midpoint of the bin.

Commonly available Bayesian Network software packages only allow hard discretization of continuous variables [8]. They propose a probability-based soft discretization solution to this problem. Soft discretization is a weighted scheme in which a value does not have to belong exclusively to one bin. For instance, the value 18 would be given a large weight in bin 2, and a smaller weight in bin 3. In this way, less of the information about where a value lies within the bin is lost. Modifying the Bayesian Model was beyond the scope of this research work. Instead we simulated soft discretization by relaxing our metric a posterori.

The use of the junction tree algorithm for efficient probability propagation in multi-connected networks was first described by Lauritzen & Spiegelhalter [4]. There are now, however, many different algorithms for this purpose, as described by Neapolitan [5]. All of these algorithms begin by moralizing the network, making it chordal, and decomposing it into a junction tree. After that, the algorithms vary as to how they actually perform the propagation of probabilities throughout the network. See, for example, Jensen & Jensen [9], Weisstein [10], and McGlohon [11] to gain a better understanding of this process.

3 Data Location and Sobek Model

The United States Geological Survey (USGS) maintains data collection stations at multiple sites along the Kootenai River, located in northern Idaho. We utilize a numerical model constructed to replicate the river flow properties at the "Tribal Hatchery" site, located approximately 150 miles upstream from the mouth of the river. The Sobek model is one dimensional, and includes momentum and continuity equations for a variety of flow dynamics including sediment load. It has been verified to be in the regions of strong sediment transfer [12]. It uses the Boussinesq approximation and includes hydraulic parameterization, as well as the standard cross sectional area flow parameters.

A typical river model might require several years to construct, and decades to calibrate. For this reason, we utilize a set of results from an existing, validated numerical riverine model of the Tribal Hatchery site. This model data was generated by the SOBEK model suite, produced by Delft Hydraulics [13]. This powerful software tool simulates one dimensional integral processes in a water drainage system – river, canal, etc [12]. The model was configured specifically for the Tribal Hatchery site mentioned above, and was run over a range of expected conditions. This produced a table of 87,840 cases, each consisting of a value for Discharge, Velocity, Width, and Depth. We simulate having set up and run the model ourselves by selecting cases from this pre-run result set. Test case selection was performed using a well seeded pseudo random number generator. This was selected because the region exhibits significantly different flow regimes episodically and we wanted to capture a wide variety of flows.

One of the earliest studies of river dynamics was based on a study of the Po river made by the Italian engineer Guglielmini in the seventeenth century. These early studies used a transport equation with simply related variables, as $Q= V^* A$ (where A is the cross sectional area, V is velocity and Q is the discharge) . Such simple equations assume idealized flow. For example, it assumes conservation of water. Yet water may be absorbed by dry ground when a river's depth increases and a dry river bank is first wetted.

4 Performance Metric: Bin Prediction Accuracy

To test the network, we give the network values for the other 3 variables (not for the variable being tested) for all 87,840 cases. We compare the bin which the network estimates the value to be in (for each case) with the bin the value was actually in. We use a modified Bin Prediction Accuracy metric as our performance criteria – the percentage of times the network accurately chooses the correct bin.

It is desirable that the matrix from which BPA is calculated has high values along the principal diagonal, indicating that the Bayesian Network predicted the correct bin. In a "strict" BPA calculation, only the values along that principal diagonal would be counted as correct. However, Ebert-Uphoff [8] discusses the fact that, in a hard discretization scheme (such as ours), intra-bin information is lost. Thus, for values that fall near the edge of a bin, there is an increased likelihood that the Bayesian Network will predict the bin next to the "correct" bin. Figure 1 clearly shows that there are many bins with high prediction counts that are next to the actual bins the values fall into. If we only counted the actual "correct" bins as correct, we would be ignoring the distributional information inherent in the matrix – we would be counting those cases the same as when the network's prediction was completely wrong. Thus, we include those bins as "correct" in our BPA calculations. This soft correction is an *a posteriori* accommodation, in contrast to Ebert-Uphoff's [8] *a priori* soft discretization, intended to accomplish the same goal, *albeit* less rigorously.

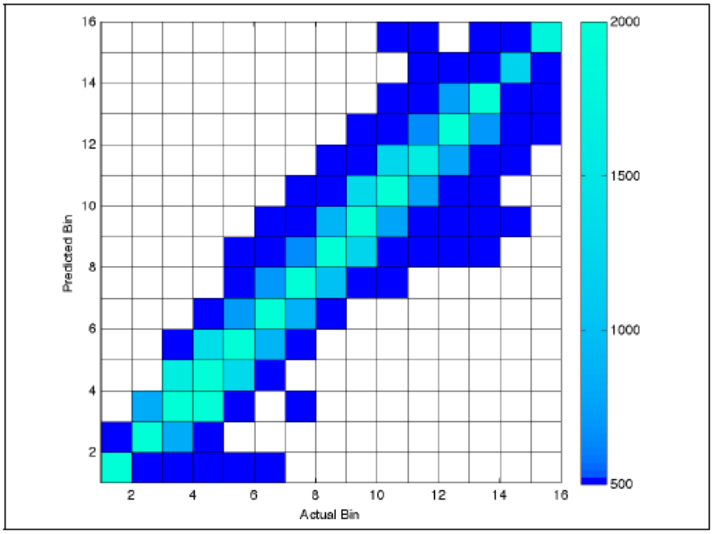


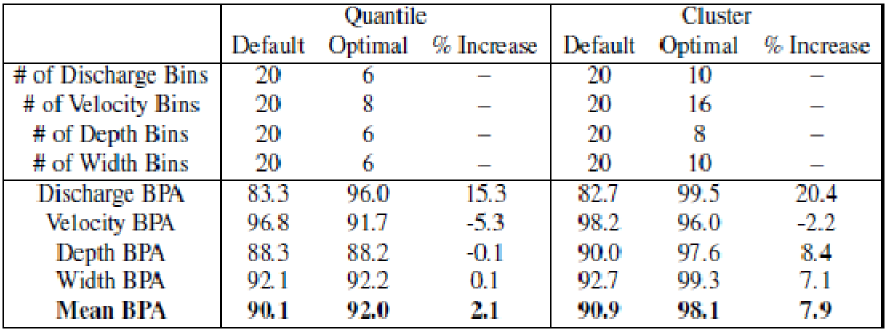
Fig. 1. Bin Prediction Accuracy: Predicted Bin Vs Actual Bin

5 Discretization Optimization

Discretization involves answering two questions: Where should the bin boundaries ('breaks') be placed? How many bins should there be? Optimal solutions to these questions are computationally intensive and often not intuitive. In Netica (and other Bayesian Network software packages), the default method of answering the first question is to use a quantile method to place the breaks such that an even number of cases falls into each bin. There is no default answer to the second question; researchers generally must explicitly choose the number of bins. In this paper, we develop an algorithm to answer both of these questions in a way that improves the overall accuracy of the resulting network. In the quantile method, bin boundaries are placed such that an even number of cases falls into each bin. Essentially, it orders the cases, then splits them into evenly spaced groups. For instance, if we were using 20 bins, the first 5% would go into bin 1, the next 5% into bin 2, and so on.

It seems intuitive that placing bins such that an even number of cases fall into each bin would yield high BPA. The problem, however, lies in the fact that the bin boundaries are placed at pre-determined places in the Cumulative Distribution Function. So bin boundaries can (and often do) get placed in the middle of a dense cluster of cases that would preferably be grouped together in one bin because they represent similar dynamics. A data clustering approach addresses this bin boundary placement directly. This method locates these dense clusters of cases, and places the bin boundaries such that the clusters are binned together, forming natural breaks. As evidenced in table 1, the optimal bin count combination using this method yields approximately 8% higher Bin Prediction Accuracies than the optimal bin count combination using the quantile method.

Table 1. Discretization Optimization Results



	Quantile			Cluster		
	Default	Optimal	% Increase	Default	Optimal	% Increase
# of Discharge Bins	20	6	–	20	10	–
# of Velocity Bins	20	8	–	20	16	–
# of Depth Bins	20	6	–	20	8	–
# of Width Bins	20	6	–	20	10	–
Discharge BPA	83.3	96.0	15.3	82.7	99.5	20.4
Velocity BPA	96.8	91.7	-5.3	98.2	96.0	-2.2
Depth BPA	88.3	88.2	-0.1	90.0	97.6	8.4
Width BPA	92.1	92.2	0.1	92.7	99.3	7.1
Mean BPA	**90.1**	**92.0**	**2.1**	**90.9**	**98.1**	**7.9**

For verification testing, we use a brute-force approach to determine the optimal number of bins for each variable. We test all even numbers between 6 and 20 as bin counts for each variable, using both methods (quantile and data clustering) mentioned above. For each bin count combination, we record the BPA for each variable. For each bin boundary placement method, we choose the bin count combination that yields the highest mean BPA. Discretization optimization leads to increased prediction accuracy —-bins matter. But discretization also yields another benefit. Because it may lead to the use of fewer bins per variable, discretization optimization may yield faster computing times, since there are less bins across which the junction tree algorithm must propagate probabilities. Using all 87,840 cases from the SOBEK model output, the Bayesian Network is able to predict a single unknown variable with 98.1% accuracy, given values for the other three variables, as shown in table 1. Here, however, we show that the Bayesian Network can reach nearly the same prediction accuracy levels, using far fewer than the entirety of the 87,840 cases. In our study a random selection of only 1,000 cases resulted in accuracy above 90% (Fig 2). The network's ability to accurately predict which bin a variable's value will fall into (1) quickly reaches nearly the same accuracy levels as using all 87,840 cases, and (2) quickly shows diminishing returns – to add more data would be of decreasing benefit.

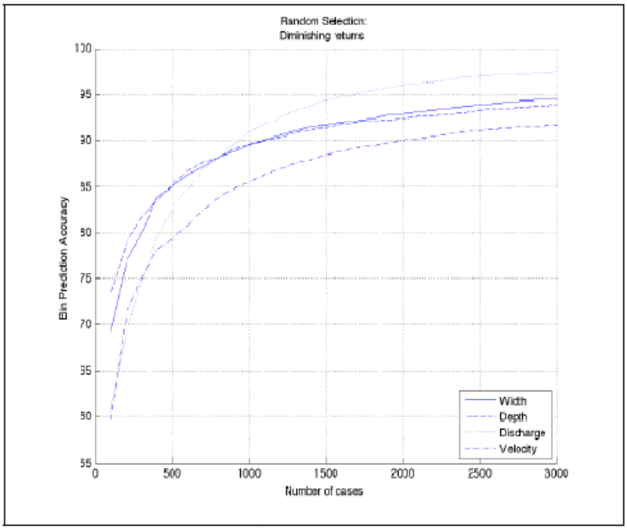


Fig. 2. Bin Prediction Accuracy Vs Number of Cases

6 Future Development

Our study showed that the network can achieve essentially the same BPA using far fewer than the full set of 87,840 cases. Thus, there should be an informed way to intelligently choose data points from the full set, rather than using random selection —- an uninformed method. A consequence of discretizing the SOBEK model data is a loss of information. For instance, consider a situation in which bin 3 of variable V contains all values in the range 3 to 5. In this situation, the difference between a value of 3.1 and a value of 4.9 has been lost, since both fall into bin 3. It seems likely that the network's BPA would benefit from the use of soft discretization [8]. This is a weighted discretization scheme in which it is not necessary to assign all of a value's weight to only one bin. For instance, the value of 3.1 might have a large weight assigned to bin 3, and a smaller weight assigned to bin 2, indicating that it was near the lower bound of bin 3. In this way, we would save some of the information that is now being lost due to the hard discretization scheme. However, as [8] discusses, there are currently no Bayesian Network software packages that allow soft discretization. Thus, a Bayesian Network software package would have to be modified to allow this.

7 Conclusion

As mentioned in the section describing BPA, we include both the bins next to and on the principal diagonal in Fig.1 as correct. It is likely that there is a more optimal way to include these bins in our calculations. Perhaps an exponentially decreasing weight would be more appropriate. In this scheme, the bins along the principal diagonal would be counted as correct with a weight of 1. Bins that are one bin away from the principal diagonal would be counted, but with a smaller weight, and bins that are two bins away would be counted, but with a still smaller weight. More work is needed to determine the coefficient weights for an exponentially decaying scheme.

The discretization improvements to Bayesian network forecast methods are general in nature, and may be applied to many other numerical models. The work is not limited to our riverine model. Netica's default discretization scheme is quantile, usually using the same number of bins for every variable. Using this scheme (with twenty bins for each of our four variables), the network achieved 90.1% accuracy. We developed a discretization optimization algorithm to select a discretization scheme that yields a network that is both more accurate and more computationally efficient. This algorithm uses data clustering to find natural breaks between groups of clustered data. By changing the discretization scheme and intelligently selecting the number of bins for each variable, we increased the network's accuracy from 90.1% to 98.1%.

It is important to note that the value of these contributions is not limited to the application of Bayesian Networks in a riverine system. Rather, the concepts are general enough that they can be applied to any system in which the data must be discretized before it can be used in a Bayesian Network. Thus, they should be of benefit not only to geoscientists, but also to physicists, engineers, economists, and researchers in any discipline to which numerical modeling can be applied.

We modified our accuracy measure as well. It is an improvement over a strict bin prediction accuracy metric in that it maintains not only cases in which the network predicted the correct bin, but also cases in which the network was only off by one bin. In a stricter metric, these cases would be counted equally as wrong as ones for which the network predicted a completely wrong bin. This metric is also an improvement over the standard hindcasting method of determining a network's accuracy. In hindcasting, the actual value is compared with the mean of the bin predicted by the network. So, for all cases in which the value does not fall exactly in the center of the bin, the network is penalized as having "error", even though (since it is discretized) it cannot perform any better than predicting the correct bin. In short, hindcasting attempts to extract a continuous value from a discrete variable. Our metric improves on this by only trying to extract a discrete value from a discrete variable. Because the network can do no better than predicting the correct bin, that is counted as completely correct in our modified bin prediction accuracy metric.

Finally, we showed that the network's accuracy exhibits diminishing returns as more data is added. This should be a significant help to researchers to help them determine how much data needs to be collected from laboratory and field experiments in order to achieve results that are at least within some pre-defined accuracy threshold. Further, the iterative learning algorithm we developed can be used (in conjunction with a numerical model) to determine, a priori, approximately how many data points must be collected to achieve a desired accuracy level. This can save researchers much valuable time, money, and resources that could otherwise be wasted on collecting unnecessary data.

References

1. Agena Bayes Theorem, `http://www.agenarisk.com/resources/probability_puzzles/bayes_theorem.shtml`
2. Niedermayer, D.: An Introduction to Bayesian Networks and their Contemporary Applications (December 1998), `http://www.niedermayer.ca/papers/bayesian/index.html.old`
3. Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach, 3rd edn. Pearson Education (2009)
4. Lauritzen, S.L., Spiegelhalter, D.J.: Local Computations with Probabilities on Graphical Structures and their Application to Expert Systems. Journal of the Royal Statistical Society 50(2), 157–224 (1988)
5. Neapolitan, R.E.: Learning Bayesian Networks. In: The 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2007, p. 5:1. ACM, New York (2007)
6. Fenton, N., Neil, M.: Managing Risk in the Modern World: Applications of Bayesian Networks, Technical report, London Mathematical Society and Knowledge Transfer Network for Industrial Mathematics (2007)
7. Norsys Software Corporation: Netica Bayesian Network Software (2011), `http://www.norsys.com`
8. Ebert-Uphoff, I.: A Probability-Based Approach to Soft Discretization for Bayesian Networks, Technical report, Georgia Institute of Technology (2009)
9. Jensen, F.V., Jensen, F.: Optimal Junction Trees. In: Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence, pp. 360–366. Morgan Kaufmann Publishers Inc., San Francisco (1994)
10. Weisstein, E.W.: Chordal Graph, From MathWorld–A Wolfram Web Resource, `http://mathworld.wolfram.com/ChordalGraph.html`
11. McGlohon, M.: Methods and Uses of Graph Demoralization, `http://www.cs.cmu.edu/~mmcgloho/pubs/sigbovik3.pdf`
12. Ji, Z., de Vriend, H., Hu, C.: Application of Sobek Model in the Yellow River Estuary. In: International Conference on Estuaries and Coasts, Hangzhou, China (November 2003)
13. Delft Hydraulics Software: SOBEK Suite (2011), `http://delftsoftware.wldelft.nl/index.php?task=blogcategory&Itemid=35`

Application of General Type-2 Fuzzy Set in Emotion Recognition from Facial Expression

Anisha Halder[1], Rajshree Mandal[2], Aruna Chakraborty[2],
Amit Konar[1], and Ramadoss Janarthanan[3]

[1] Department of Electronics and Tele-Communication Engineering,
Jadavpur University, Kolkata-32, India
[2] Department of Computer Science and Engineering,
St. Thomas' College of Engineering and Technology, Kolkata, India
[3] Department of IT, Jaya Engg. College, Chennai
`{halder.anisha,rajshree.mondal}@gmail.com,`
`aruna_stcet@rediffmail.com, konaramit@yahoo.co.in,`
`srmjana_73@yahoo.com`

Abstract. This paper proposes a new technique for emotion recognition of an unknown subject using General Type-2 Fuzzy sets (GT2FS). The proposed technique includes two steps- first, a type-2 fuzzy face-space is created with the background knowledge of facial features of different subjects containing different emotions. Second, the emotion of an unknown facial expression is determined based on the consensus of the measured facial features with the fuzzy face-space. The GT2FS has been used here to model the fuzzy face space. The general type-2 fuzzy involves both primary and secondary membership distributions which have been obtained here by formulating and solving an optimization problem. The optimization problem here attempts to minimize the difference between two decoded signals: the first one being the type-1 defuzzification of the average primary membership distributions obtained from n-subjects, while the second one refers to the type-2 defuzzified signal for a given primary distribution with secondary memberships as unknown. The uncertainty management policy adopted using general type-2 fuzzy set has resulted in a classification accuracy of 96.67%.

Keywords: Emotion Recognition, Facial feature extraction, Type-2 primary membership, Type-2 secondary membership, Fuzzy Face Space.

1 Introduction

Emotion recognition is currently gaining importance for its increasing scope of applications in human-computer interactive systems. Several modalities of emotion recognition, including facial expression, voice, gesture and posture have been studied in the literature. However, irrespective of the modality, emotion recognition comprises two fundamental steps involving feature extraction and classification. Feature extraction refers to determining a set of features/attributes, preferably independent, which together represents a given emotional expression. Classification aims at mapping emotional features into one of several emotion classes.

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 460–468, 2011.
© Springer-Verlag Berlin Heidelberg 2011

Among the well-known methods of determining human emotions, Fourier descriptor [1], template matching [2], and neural network techniques [3], [4] deserve special mention. Other important works undertaken so far for recognition of emotions from facial expression by selecting suitable features include [5], [6], [7] and by identifying the right classifier include [3], [4],[10].

The paper provides an alternative approach to emotion recognition from an unknown facial expression, when the emotion class of individual facial expression of a large number of experimental subjects is available. The General Type-2 fuzzy set (GT2FS) based approach employs to construct a fuzzy face space, comprising both primary and secondary membership distributions, obtained from known facial expressions of several subjects containing multiple instances of the same emotion for each subject. The emotion class of an unknown facial expression is determined by obtaining maximum support of each class to the given facial expression. The class with the maximum support is the winner. The maximum support evaluation here employs both primary and secondary distributions. Experiments reveal that the classification accuracy of emotion of an unknown person by the GT2FS based scheme is as high as 96%.

The paper is divided into five sections. In section 2, we propose the principle of GT2FS approach. Methodology of the proposed scheme is discussed in section 3. Experimental details are given in section 4. Conclusions are listed in section 5.

2 Principles Used in the GT2FS Approach

The GT2FS based reasoning realized with measurements taken from n-subjects, requires k × m × n general type-2 fuzzy sets to determine the emotion class of an unknown facial expression where, k is the number of emotion classes and m is the number of features.

Let $F=\{f_1, f_2, \ldots, f_m\}$ be the set of m facial features. Let $\mu_{\tilde{A}}(f_i)$ be the primary membership in [0,1] of the feature f_i to be a member of set $\tilde{A}$, and $\mu(f_i, \mu_{\tilde{A}}(f_i))$ be the secondary membership of the measured variable f_i in [0,1]. If the measurement of a facial feature, f_i, is performed p times on the same subject experiencing the same emotion, and the measurements are quantized into q intervals of equal size, we can evaluate the frequency of occurrence of the measured variable f_i in q quantized intervals. The interval containing the highest frequency of occurrence then can be identified, and its centre, m_i, approximately represents the mode of the measurement variable f_i. The second moment, σ_i, around m_i is determined, and an exponential bell-shaped (Gaussian) membership function centered on m_i and with a spread σ_i is used to represent the membership function of the random variable f_i. This function represents the membership of f_i to be CLOSE-TO the central value, m_i. It may be noted that a bell-shaped (Gaussian-like) membership curve would have a peak at the centre with a membership value one, indicating that membership at this point is the largest for an obvious reason of having the highest frequency of f_i at the centre.

On repetition of the above experiment for variable f_i on n subjects, each experiencing the same emotion, we obtain n such membership functions, each one for one individual subject. Naturally, the measurement variable f_i now has both intra- and inter-personal level uncertainty. The intra-level uncertainty occurs due to the pre-assumption of the bell-shape (Gaussian distribution) of the primary membership function, and the inter-level uncertainty occurs due to multiplicity of the membership distribution for n subjects. Thus a new measurement for an unknown facial expression can be encoded using all the n-membership curves, giving n possible membership values, thereby giving rise to uncertainty in the fuzzy space.

GT2FS approach tunes the primary membership values for the given measurements using the secondary memberships of the same measurement, and thus reduces the degree of intra-level uncertainty of the primary distributions. The reduction in the degree of uncertainty helps in improving the classification accuracy of emotion at the cost of additional complexity required to evaluate type-2 secondary distributions and also to reason with k × m × n fuzzy sets.

Let $fi^{/}$ be the measurement of the i-th feature for an unknown subject. Now, by consulting the n primary membership distributions obtained from n-subjects for a given emotion, we obtain n primary membership values for the measurement $f_i^{/}$ given by $\mu_{\tilde{A}}^{1}(f_i^{/}), \mu_{\tilde{A}}^{2}(f_i^{/}), \ldots\ldots, \mu_{\tilde{A}}^{n}(f_i^{/})$. Let the secondary membership values for each primary membership respectively be $\mu(f_i^{/}, \mu_{\tilde{A}}^{1}(f_i^{/})), \mu(f_i^{/}, \mu_{\tilde{A}}^{2}(f_i^{/})), \ldots\ldots, \mu(f_i^{/}, \mu_{\tilde{A}}^{n}(f_i^{/}))$. Since the secondary memberships denote the degree of accuracy of the primary memberships, the uncertainty in a primary membership distribution can be reduced by multiplying each primary membership value by its secondary membership. Thus the modified primary membership values are given by

$$\left.\begin{array}{l} {}^{\text{mod}}\mu_{\tilde{A}}^{1}(f_i^{/}) = \mu_{\tilde{A}}^{1}(f_i^{/}) \times \mu(f_i^{/}, \mu_{\tilde{A}}^{1}(f_i^{/})), \\ {}^{\text{mod}}\mu_{\tilde{A}}^{2}(f_i^{/}) = \mu_{\tilde{A}}^{2}(f_i^{/}) \times \mu(f_i^{/}, \mu_{\tilde{A}}^{2}(f_i^{/})), \\ \ldots\ldots\ldots\ldots\ldots\ldots\ \ldots\ \ldots\ldots\ldots\ \ldots\ \ldots \\ {}^{\text{mod}}\mu_{\tilde{A}}^{n}(f_i^{/}) = \mu_{\tilde{A}}^{n}(f_i^{/}) \times \mu(f_i^{/}, \mu_{\tilde{A}}^{n}(f_i^{/})) \end{array}\right\} \tag{1}$$

where ${}^{\text{mod}}\mu_{\tilde{A}}^{j}(f_i^{/})$ denotes the modified primary membership value for j-th subject. The next step is to determine the range of ${}^{\text{mod}}\mu_{\tilde{A}}^{j}(f_i^{/})$ for j= 1 to n, comprising the minimum and the maximum given by $[{}^{\text{mod}}\underline{\mu}_{\tilde{A}}(f_i^{/}), {}^{\text{mod}}\overline{\mu}_{\tilde{A}}(f_i^{/})]$, where

$${}^{\text{mod}}\underline{\mu}_{\tilde{A}}(f_i^{/}) = Min\ \{{}^{\text{mod}}\mu_{\tilde{A}}^{1}(f_i^{/}), {}^{\text{mod}}\mu_{\tilde{A}}^{2}(f_i^{/}), \ldots, {}^{\text{mod}}\mu_{\tilde{A}}^{n}(f_i^{/})\},$$

$${}^{\text{mod}}\overline{\mu}_{\tilde{A}}(f_i^{/}) = Max\ \{{}^{\text{mod}}\mu_{\tilde{A}}^{1}(f_i^{/}), {}^{\text{mod}}\mu_{\tilde{A}}^{2}(f_i^{/}), \ldots, {}^{\text{mod}}\mu_{\tilde{A}}^{n}(f_i^{/})\},$$

If there exist m different facial features, then for each feature we would have such an interval, and consequently we obtain m such intervals given by

$$[{}^{\text{mod}}\underline{\mu}_{\tilde{A}}(f_1^{/}), {}^{\text{mod}}\overline{\mu}_{\tilde{A}}(f_1^{/})], [{}^{\text{mod}}\underline{\mu}_{\tilde{A}}(f_2^{/}), {}^{\text{mod}}\overline{\mu}_{\tilde{A}}(f_2^{/})], \ldots [{}^{\text{mod}}\underline{\mu}_{\tilde{A}}(f_m^{/}), {}^{\text{mod}}\overline{\mu}_{\tilde{A}}(f_m^{/})]$$

The proposed reasoning system employs a particular format of rules, commonly used in fuzzy classification problems. Consider for instance a fuzzy rule, given by

R_c: if f_1 is $\tilde{A}_1$ AND f_2 is $\tilde{A}_2$ …. AND f_m is $\tilde{A}_m$ then emotion class is c.

Here, f_i for i=1 to m are m-measurements (feature value) in the general type-2 fuzzy sets $\tilde{A}_1$, $\tilde{A}_2$, …, $\tilde{A}_m$ respectively, given by

$$\tilde{A}_i = [\underline{\mu}_{\tilde{A}}(f_i), \overline{\mu}_{\tilde{A}}(f_i)], \forall i \cdot \tag{2}$$

Since an emotion is characterized by all of these m features, to find the overall support of the m features (m measurements made for the unknown subject) to the emotion class c represented by the n primary memberships, we use the fuzzy meet operation

$$S_c^{\min} = Min\{ {}^{\text{mod}}\underline{\mu}_{\tilde{A}}(f_1^{/}), {}^{\text{mod}}\underline{\mu}_{\tilde{A}}(f_2^{/}), \ldots\ldots, {}^{\text{mod}}\underline{\mu}_{\tilde{A}}(f_m^{/}), \} \tag{3}$$

$$S_c^{\max} = Min\{ {}^{\text{mod}}\overline{\mu}_{\tilde{A}}(f_1^{/}), {}^{\text{mod}}\overline{\mu}_{\tilde{A}}(f_2^{/}), \ldots\ldots, {}^{\text{mod}}\overline{\mu}_{\tilde{A}}(f_m^{/})\} \tag{4}$$

Thus we can say that the unknown subject is experiencing the emotion class c at least to the extent s_c^{min}, and at most to the extent s_c^{max}.

To reduce the non-specificity associated with the interval S_{c-i}, the most conservative approach would be to use lower bound, while the most liberal view would to use the upper bound of the interval as the support for the class c. In absence of any additional information, a balanced approach would be to use center of the interval as the support for the class c by the n primary memberships to the unknown subject [8]. We compute the centre, S_c of the interval S_{c-i}.Thus S_c is the degree of support that the unknown facial expression is in emotion class *c*.

$$S_c = (s_c^{min} + s_c^{max})/2. \tag{5}$$

Now to predict the emotion of a person from his facial expression, we determine S_c for each emotion class. Presuming that there exist k emotion classes, let us denote them by S_1, S_2,...., S_k for emotion class *1, 2,..., k*, respectively. Since a given facial expression may convey different emotions with different degrees, we resolve the conflict by ranking the S_i for $i = 1$ *to k*, and thus determine the emotion class r, for which $S_r >= S_i$ for all *i* following the Rule R_c.

To make the algorithm robust, we consider association of fuzzy encoded measurements with emotion class by considering the weakest reliability of the joint occurrence of the fuzzy measurements, and identify the winning emotion class having this measure of reliability superseding the same of other emotion classes.

3 Methodology

We now briefly discuss the main steps involved in fuzzy face space construction based on the measurements of m facial features for n-subjects, each having *l* instances of facial expression for a particular emotion. We need to classify a facial expression of an unknown person into one of k emotion classes.

1. We extract m facial features for n subjects, each having *l* instances of facial expression for a particular emotion. The above features are extracted for k-emotion classes.
2. We construct a fuzzy face space for each emotion class separately. The fuzzy face space for an emotion class comprises a set of n primary membership and secondary membership distributions for each feature. Thus we have m groups of n-primary as well as secondary membership distributions. Each membership curve is constructed from *l*-facial instances of a subject attempted to exhibit a particular emotion in her facial expression by acting.
3. For a given feature $f_i^{/}$, we consult each primary and secondary membership curve under a given emotion class, and take the product of primary and secondary membership at $f_i = f_i^{/}$. The resulting membership value obtained for the membership curves for the subject w is given by

   $$ {}^{\mathrm{mod}}\mu_{\tilde{A}}^{w}(f_i^{/}) = \mu_{\tilde{A}}^{w}(f_i^{/}) \times \mu(f_i^{/}, \mu_{\tilde{A}}^{w}(f_i^{/})) \tag{6} $$

   Now, for w = 1 to n, we evaluate ${}^{\mathrm{mod}}\mu_{\tilde{A}}^{w}(f_i^{/})$, and thus obtain the minimum and the maximum values of ${}^{\mathrm{mod}}\mu_{\tilde{A}}^{w}(f_i^{/})$, to obtain a range of uncertainty $[{}^{\mathrm{mod}}\underline{\mu}_{\tilde{A}}(f_i^{/}), {}^{\mathrm{mod}}\overline{\mu}_{\tilde{A}}(f_i^{/})]$. This is repeated for all features under each emotion class.
4. Now for an emotion class j, we take fuzzy meet operation over the ranges for each feature to evaluate the range of uncertainty for individual emotion class. The meet operation here is computed by taking cumulative t-norm of ${}^{\mathrm{mod}}\underline{\mu}_{\tilde{A}}(f_i^{/})$ and ${}^{\mathrm{mod}}\overline{\mu}_{\tilde{A}}(f_i^{/})$ separately for i= 1 to m, and thus obtaining S_j^{min} and S_j^{max} respectively.
5. The support of the j-th emotion class to the measurements is evaluated by taking average of S_j^{min} and S_j^{max}, and defining the result by S_j.
6. Now by using classifier rule, we determine the maximum support offered by all the k emotion classes, and declare the unknown facial expression to have emotion r, if $S_r > S_i$ for all emotion class $i = 1$ *to* k.

4 Experimental Details

In this section, we present the experimental details of emotion recognition using the principles introduced in section 2 and 3. We here consider 5 emotion classes, (i.e., k=5) including anger, fear, disgust, happiness and relaxation. The experiment is conducted with two sets of subjects: a) the first set of n (=10) subjects is considered for designing the fuzzy face-space and, b) the other set of 30 facial expressions taken from 6 unknown subjects are considered to validate the result of the proposed emotion classification scheme. Five facial features, (i.e., m=5) have been used here to design the type-2 fuzzy face-space. We now briefly outline the main steps.

4.1 Feature Extraction

Feature extraction is a fundamental step in emotion recognition. Existing research results [9],[10] reveal that the most important facial regions responsible for the manifestation of emotion are the eyes and the lips. This motivated us to select the following features: Left Eye Opening (EO_L), Right Eye Opening (EO_R), Distance between the Lower Eyelid to the Eyebrow for the Left Eye (LEE_L), Distance between the Lower Eyelid to Eyebrow for the Right Eye (LEE_R), and the Maximum Mouth opening (MO) including the lower and the upper lips. Fig. 1 explains the above facial features on a selected facial image.

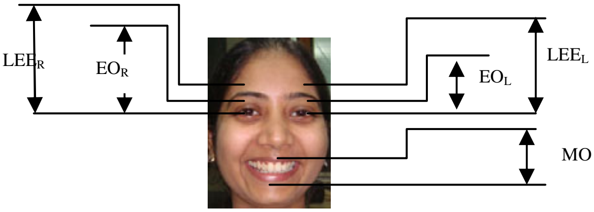


Fig. 1. Facial Features

4.2 Creating the Type-2 Fuzzy Face-Space

The Type-2 fuzzy face-space contains the primary and corresponding secondary membership distributions for each facial feature. Since we have 5 facial features and the experiment includes 5 distinct emotions of 10 subjects, we obtain 10×5×5=250 primary as well as secondary membership curves. To compute primary memberships, 10 instances of a given emotion is used. These 250 membership curves are grouped into 25 heads, each containing 10 membership curves of ten subjects for a specific feature for a given emotion. Fig. 2. (a) gives an illustration of one such group of 10 membership distributions for the feature EO_L for the emotion: disgust.

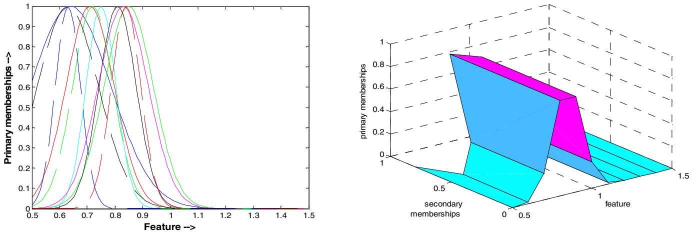


Fig. 2. (a). Membership distributions for emotion disgust and feature EO_L (b) Secondary Membership curve of Subject1

For each primary membership distribution, we have a corresponding secondary membership distribution. Thus we obtain 250 secondary membership distributions. One illustrative type-2 secondary distributions of subject 1 for the feature EO_L for the emotion disgust are given in Fig. 2. (b) The axes in the figure represent feature (EO_L), primary and secondary membership values as indicated.

4.3 Emotion Recognition of an Unknown Facial Expression

The emotion recognition problem addressed here attempts to determine the emotion of an unknown person from her facial expression. To keep the measurements in an emotional expression normalized and free from distance variation from the camera focal plane, we construct a bounding box, covering only the face region, and the reciprocal of the diagonal of the bounding box is used as a scale factor for normalization of the measurements. The normalized features obtained from Fig.3 are enlisted in Table 1. We now briefly explain the experimental results obtained by our proposed method.

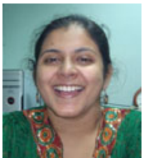


Fig. 3. Facial Image of an unknown subject

Table 1. Extracted Features of Fig. 3

EO_L	EO_R	MO	LEE_L	LEE_R
0.026	0.026	0.135	0.115	0.115

The GT2FS based recognition scheme considers a fuzzy face space of 5 sets of 10 primary membership distributions as in Fig. 2 (a), and the corresponding secondary membership distributions to the individual primary membership distribution of 5 features obtained from facial expressions carrying 5 distinct emotions for 10 different subjects are determined using curves like Fig. 2 (b).

Table 2. Calculated Type-2 Membership Values For Feature :EO_L, Emotion: Disgust

Feature	Primary Memberships (μ_{pri})	Secondary memberships (μ_{sec})	$\mu^{mod}=\mu_{pri} \times \mu_{sec}$	Range (min{ μ^{mod}},max{ μ^{mod} })
EO_L	0.65	0.72	0.468	0.044-0.468
	0.10	0.55	0.055	
	0.15	0.58	0.087	
	0.45	0.68	0.306	
	0.18	0.56	0.1008	
	0.55	0.68	0.374	
	0.08	0.55	0.044	
	0.41	0.63	0.2583	
	0.16	0.53	0.0848	
	0.12	0.59	0.0708	

Table 3. Calculated Feature Ranges and Centre Value For Each Emotion

Emotion	Range of Secondary Membership for Features					Range S_c^j after fuzzy Meet operation (**centre)**
	EO_L	EO_R	MO	LEE_L	LEE_R	
Anger	0-0.14	0 - 0.17	0.39-0.964	0.0034-0.814	0.0029-0.781	0-0.14(0.07)
Disgust	0.044-0.468	0.041-0.531	0-0	0-0.39	0.-0.283	0-0(0)
Fear	0 - 0.298	0-0.275	0.04-0.742	0.054-0.473	0.057-0.511	0-0.275(0.1375)
Happy	0 - 0.555	0-0.604	0.573-0.910	0.133-0.851	0.3214-0.7213	0-0.555(**0.2775**)
Relaxed	0 - 0.132	0-0.221	0-0	0.013-0.458	0.046-0.552	0-0(0)

Table 2 provides the summary of the primary and secondary memberships obtained for EO_L for the emotion: disgust. For each feature we obtain 5 Tables like Table 2, each one for a given emotion. Thus for 5 features, we would have altogether 25 such tables. In Table 2, we also computed the product of primary and secondary memberships, and then obtain the minimum and maximum of the product to determine its range, as indicated in the last column of Table 2.

The range for each feature corresponding to individual emotions is given in Table 3. For example, the entry (0-0.14) corresponding to the row Anger and column EO_L, gives an idea about the extent of the EO_L for the unknown subject matches with known subjects from the emotion class Anger. The results of computing fuzzy meet operation over the range of individual features taken from facial expressions of the subjects under same emotional condition are given in Table 3. The average of the ranges along with its centre value is also given in Table 3. It is observed that the centre has the largest value (=0.2775) for the emotion: happiness

5 Conclusion

The paper employs GT2FS-based automatic emotion recognition of an unknown facial expression, when the background knowledge about a large face database with known emotion class are available. The GT2FS-based recognition scheme requires type-2 secondary membership distributions, a computation of which by an evolutionary approach is also provided. The scheme first construct a fuzzy face space, and then infer the emotion class of the unknown facial expression by determining the maximum support of the individual emotion classes using the pre-constructed fuzzy face space. The class with the highest support is regarded as the emotion of the unknown facial expression. The scheme, however, takes care of both the inter- and intra-subject level uncertainty, and thus offers a higher classification accuracy for the same set of features. Experimental analysis confirms that the classification accuracy of emotion by employing GT2FS is 96.67%.

References

1. Uwechue, O.A., Pandya, S.A.: Human Face Recognition Using Third-Order Synthetic Neural Networks. Kluwer, Boston (1997)
2. Biswas, B., Mukherjee, A.K., Konar, A.: Matching of digital images using fuzzy logic. AMSE Publication 35(2), 7–11 (1995)
3. Bhavsar, A., Patel, H.M.: Facial Expression Recognition Using Neural Classifier and Fuzzy Mapping. In: IEEE Indicon 2005 Conference, Chennai, India (2005)
4. Guo, Y., Gao, H.: Emotion Recognition System in Images Based on Fuzzy Neural Network and HMM. In: Proc. 5th IEEE Int. Conf. on Cognitive Informatics (ICCI 2006). IEEE (2006)
5. Rizon, M., Karthigayan, M., Yaacob, S., Nagarajan, R.: Japanese face emotions classification using lip features. In: Geometric Modelling and Imaging (GMAI 2007), Universiti Malaysia Perlis, Jejawi, Perlis, Malaysia. IEEE (2007)
6. Kobayashi, H., Hara, F.: Measurement of the strength of six basic facial expressions by neural network. Trans. Jpn. Soc. Mech. Eng. (C) 59(567), 177–183 (1993)
7. Ekman, P., Friesen, W.V.: Unmasking the Face: A Guide to Recognizing Emotions From Facial Clues. Prentice-Hall, Englewood Cliffs (1975)
8. Mendel, J.M.: On the importance of interval sets in type-2 fuzzy logic systems. In: Proc. Joint 9th IFSA World Congress 20th NAFIPS Int. Conf., Vancouver, BC, Canada, July 25-28, pp. 1647–1652 (2001)
9. Chakraborty, A., Konar, A., Chakraborty, U.K., Chatterjee, A.: Emotion Recognition From Facial Expressions and Its Control Using Fuzzy Logic. IEEE Transactions on Systems, Man and Cybernetics (2009)
10. Das, S., Halder, A., Bhowmik, P., Chakraborty, A., Konar, A., Nagar, A.K.: Voice and Facial Expression Based Classification of Emotion Using Linear Support Vector. In: 2009 Second International Conference on Developments in eSystems Engineering. IEEE (2009)

Design of a Control System for Hydraulic Cylinders of a Sluice Gate Using a Fuzzy Sliding Algorithm

Wu-Yin Hui and Byung-Jae Choi

School of Electronic Engineering, Daegu University, Jillyang,
Gyeongsan, Gyeongbuk 712-714, Korea
huiwuyin@gmail.com, bjchoi@daegu.ac.kr

Abstract. The hydraulic system has the characteristic of high pressure, high speed, high efficiency, low noise, and high durability. Hydraulic has a wide range of applications. In this paper, we consider two control objectives of this hydraulic system. The first is to push or pull the sluice gate with an exact speed to an exact position. The second is to remove the displacement different between two cylinders when they are moving together. In order to design the controller for this system, we propose a fuzzy-sliding position control system. And to remove the difference between two cylinders, we propose the fuzzy PI synchronous controller. We show some simulation results of its availability.

Keywords: Hydraulic Cylinders, Fuzzy Logic Control, Position Controller, Synchronous Controller, Sluice Gate.

1 Introduction

The hydraulic system is one of the important mechanical systems. The hydraulic system has the characteristic of high pressure, high speed, high efficiency, low noise, and high durability. Hydraulic has a wide range of applications. For instance, the bulldozer and excavator in construction machinery, hydraulic jack in building machinery, combine harvester and tractor in agricultural machinery, and the ABS system in automobile are all examples of hydraulic system applications.

The hydraulic cylinder system has been discussed in many papers. It can convert hydraulic energy into mechanical, and it moves in straight reciprocating motion. In order to control the position of a cylinder, a servo valve can be used. Servo valve can change the supply flow by changing the area of orifice. A servo-hydraulic system can be used to provide large processing force and has a good positioning capability, but this system is complex and highly nonlinear.

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 469–476, 2011.
© Springer-Verlag Berlin Heidelberg 2011

In this study, the synchronous position control of a sluice gate with two cylinders is considered. In order to push the cylinders to an exact position with an exact speed and to remove the difference between the displacements of the two cylinders, two sliding-fuzzy controllers and two fuzzy PI controllers have been designed. We performed simulated to show the effectiveness of the proposed control system design. The result that combined and emulated the control system in Matlab –Simulink and AMESim (Advanced Modeling for Simulation of Engineering) coincided with the target.

2 System Description

In this system, which is shown Figure 1, the input of the hydraulic servo system is current. The spool of hydraulic servo valve is based on the input current. The spool movement opens the control orifice. After opening the orifice, the hydraulic can flow through the pass orifice and pass in or out of the cylinder chambers. The supply flow and the supply pressure are based on the open area of orifice depending on the current. The hydraulic oil that flows through orifice creates pressure difference between head side and load side of the piston. And this pressure difference pushes or pulls the sluicegate. Power is produced by the linear motion of the piston in cylinder.

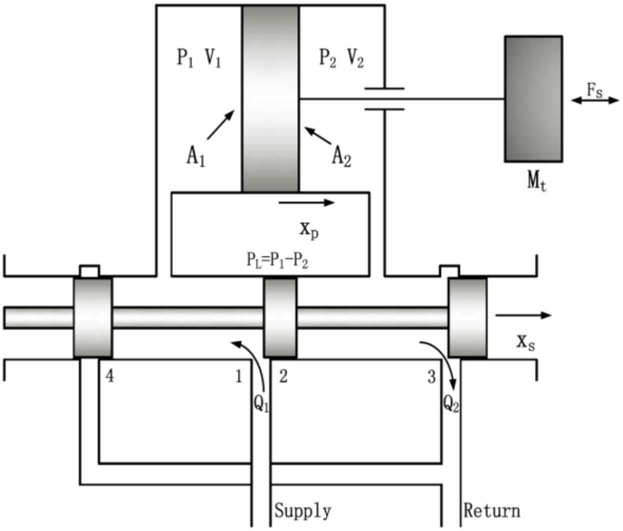


Fig. 1. Composition of hydraulic servo system

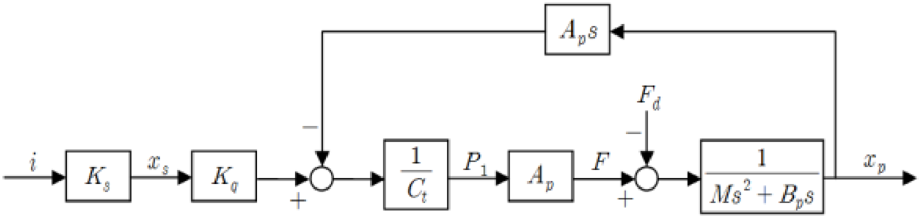


Fig. 2. The overall structure of a hydraulic servo system

3 Design of Fuzzy Sliding Controller

We consider two control objectives: the first objective is to push or pull the sluice gate with an exact speed to an exact position. The second objective is to remove the displacement difference between two cylinders when they are moving together. In order to design the controller for this system, we designed a fuzzy-sliding position control system. And a fuzzy PI synchronous controller is designed to remove the difference between two cylinders. Although there are two controllers, there is only one output that is calculated by the output values of both the position controller and synchronous controller.

3.1 Fuzzy Sliding Position Controller

In the sliding model, the control law contains both the equivalent control law and the switching law. The equivalent control law keeps the system state on the sliding surface, and the switching law can constrain the system state to move on the sliding surface. By using a fuzzy algorithm, the system contains both the equivalent control law and the switching law.

And the fuzzy membership functions are shown in Fig. 3.

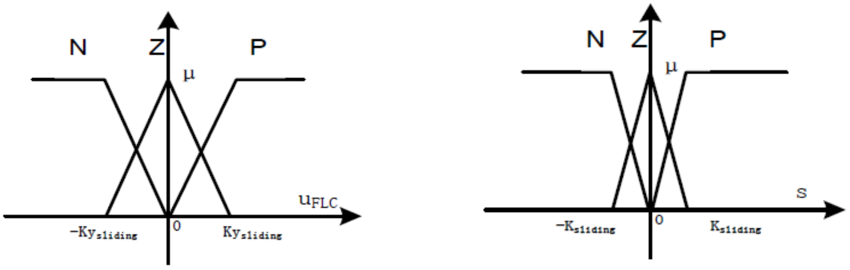
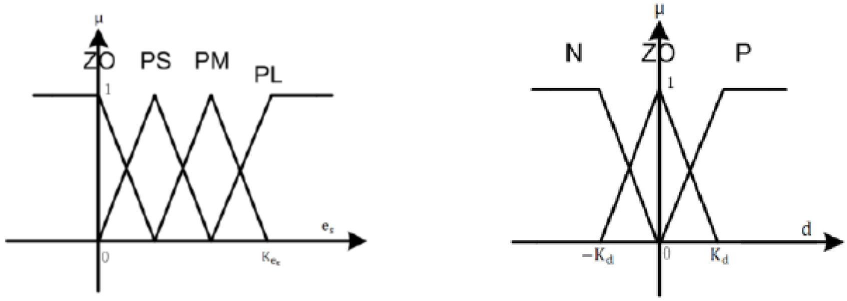


Fig. 3. The input and output membership functions for the fuzzy logic controller

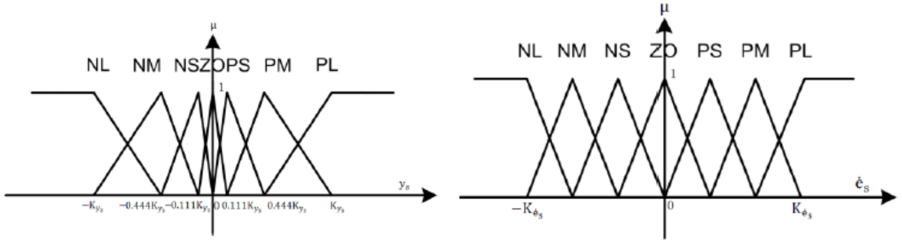
3.2 Fuzzy PI Synchronous Controller

Another controller is used to remove the displacement difference between the two cylinders. However, some problems can take place when we use two controllers together. In this paper, the synchronous controller is designed with three inputs, the first and the second ones are the difference between two cylinders and its differential coefficient, while the last one is the gap between the reference signal and displacement signal of each cylinder. In this case, when the displacement signal is close to the reference signal, the output of synchronous controller will be reduced. Moreover, even when the cylinders displacement is larger or equal to the reference signal, the output of the fuzzy PI synchronous controller is ZO. The membership functions of the synchronous controller are in following figure and the rules are shown in following table.



(a) Input "e"

(b) Input "d"

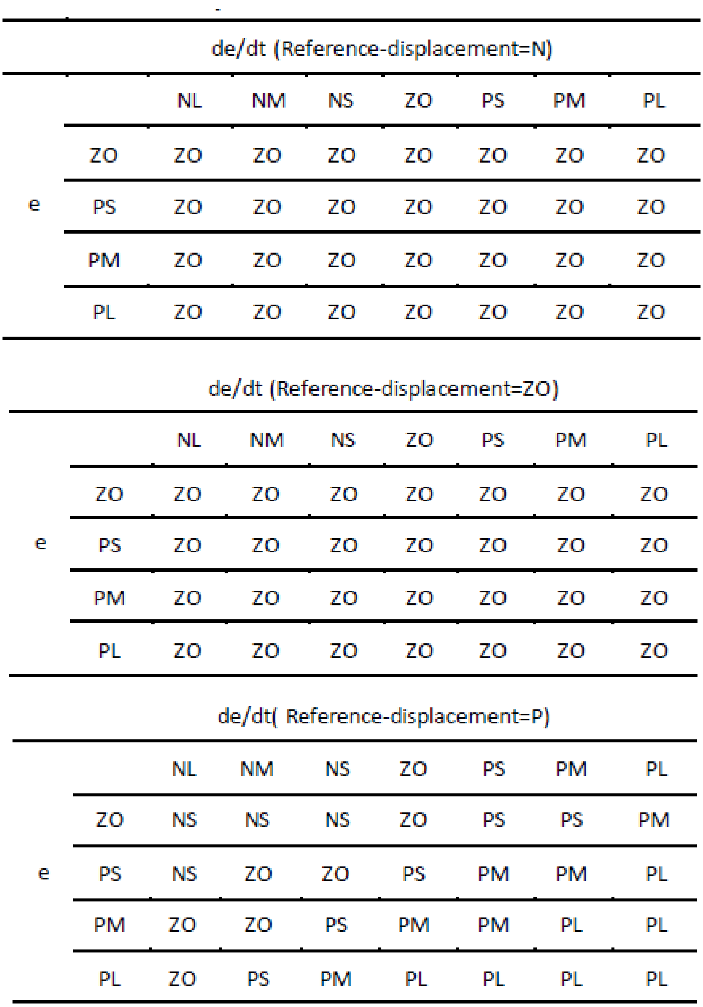


(c) Input "de/dt"

(d) Output

Fig. 4. Membership functions of the input for the synchronous controller

Table 1. Rule table for synchronous controller



de/dt (Reference-displacement=N)								
		NL	NM	NS	ZO	PS	PM	PL
e	ZO	ZO	ZO	ZO	ZO	ZO	ZO	ZO
	PS	ZO	ZO	ZO	ZO	ZO	ZO	ZO
	PM	ZO	ZO	ZO	ZO	ZO	ZO	ZO
	PL	ZO	ZO	ZO	ZO	ZO	ZO	ZO

de/dt (Reference-displacement=ZO)								
		NL	NM	NS	ZO	PS	PM	PL
e	ZO	ZO	ZO	ZO	ZO	ZO	ZO	ZO
	PS	ZO	ZO	ZO	ZO	ZO	ZO	ZO
	PM	ZO	ZO	ZO	ZO	ZO	ZO	ZO
	PL	ZO	ZO	ZO	ZO	ZO	ZO	ZO

de/dt(Reference-displacement=P)								
		NL	NM	NS	ZO	PS	PM	PL
e	ZO	NS	NS	NS	ZO	PS	PS	PM
	PS	NS	ZO	ZO	PS	PM	PM	PL
	PM	ZO	ZO	PS	PM	PM	PL	PL
	PL	ZO	PS	PM	PL	PL	PL	PL

3.3 Control System Output Algorithm

In this paper, two controllers have been designed for the control system. The output is calculated using output values of both position controller and synchronous controller. The control system is shown in following figure.

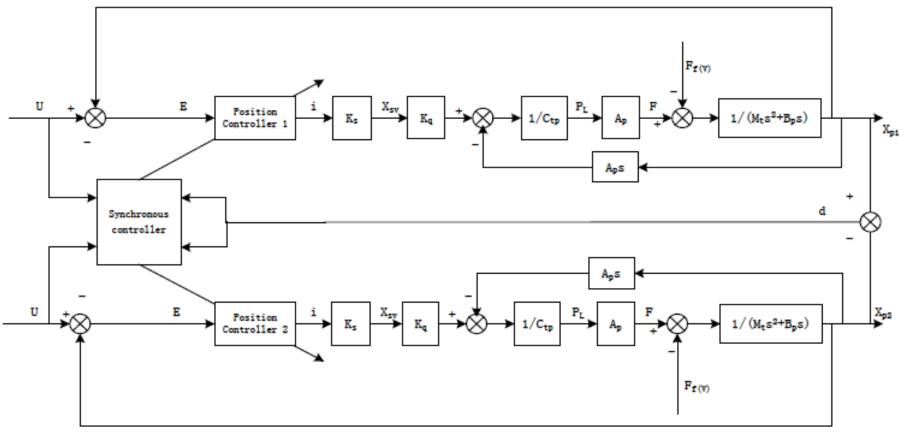


Fig. 5. The system structure of PID control system

4 Simulation Results

The system has been combined and emulated in both Matlab –Simulink and AMSim. The controller was modeled by Matlab –Simulink, and the control plant was constructed in AMSim. The overall strure is shown in following Figure.

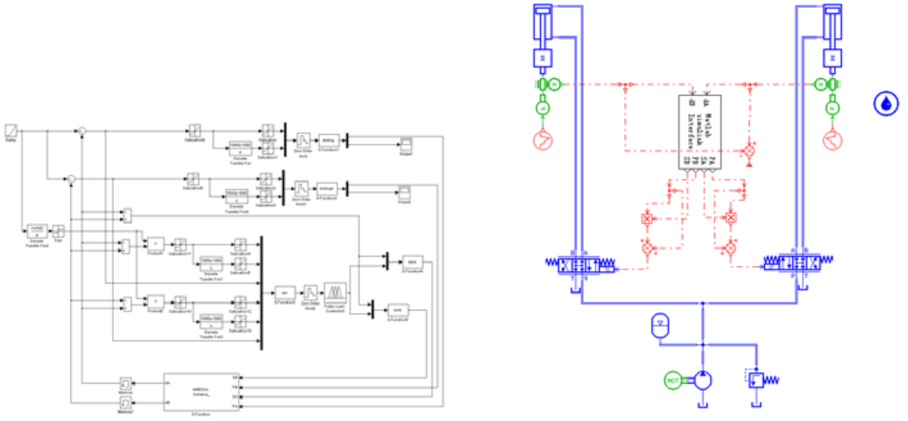


Fig. 6. The model of the overall control system and controlled process

4.1 Simulation for Extending the Stroke of 120 Seconds

The speed of cylinders is set to 4.7mm/s. the simulation results of the output of controllers are shown in following figure.

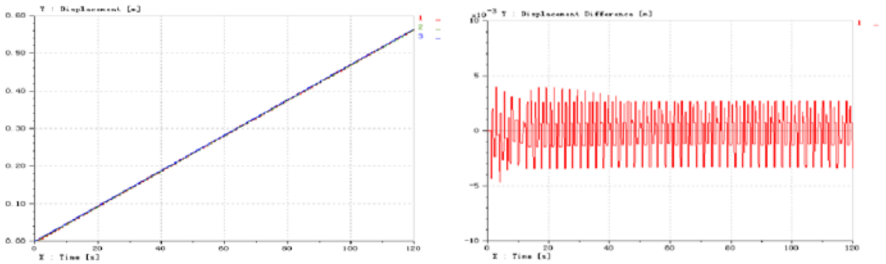


Fig. 7. Displacement for cylinder A and B, and its difference (at the extending stroke)

4.2 Simulation for Retracting the Stroke of 120 Seconds

The speed of cylinders is set to -2.3mm/s. the simulation results of the output of controllers are shown in following figure.

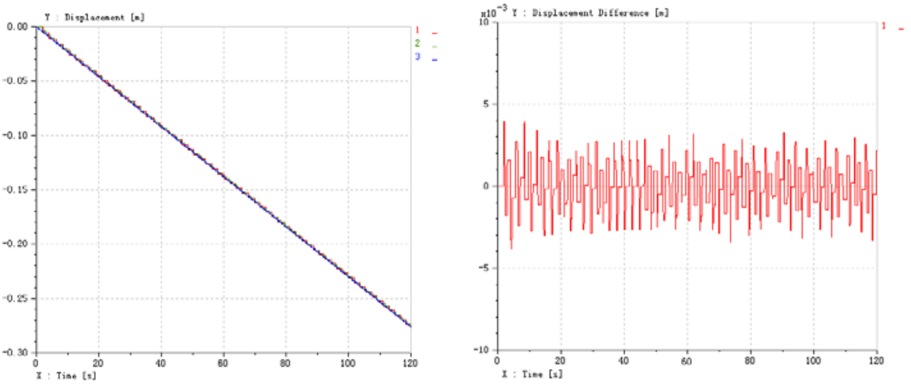


Fig. 8. Displacement for cylinder A and B, and its difference (at the retracting stroke)

5 Conclusion

In this paper, the fuzzy sliding position controller and fuzzy PI synchronous controller have been considered. The synchronous controller could regulate the output based on the difference between the reference signal and the position signal. Form the simulation result, the cylinders could track with the reference signal correctly. The difference of cylinders was bound well.

References

1. Zhao, Y.-X., Chen, X.-D., Chen, X.: Repeatability Analysis of a Vulcanizer Hydraulider System using Fuzzy Arithmetic. In: International Conference on Computational Intelligence for Measurement System and Application

2. Kim, J.H.: A Suggestion of Nonlinear Fuzzy PID Controller to Improve Transient Responses of Nonlinear or Uncertain System. International Journal of Fuzzy Logic and Intelligent System 5(4), 87–100 (1995)
3. Lee, C.C.: Fuzzy Logic in Control System. IEEE Trans. On System, Man and Cybernetica 20(2) (1990)
4. Hung, H.T., Kwan, A.K.: A Study on the Position Control of Hydraulin Cylinder Driven by Hydraulic Transformer using Disturbance Observer. In: International Conference on Control, Automation and Systems (2008)

Rough Sets for Selection of Functionally Diverse Genes from Microarray Data

Sushmita Paul and Pradipta Maji

Machine Intelligence Unit, Indian Statistical Institute, Kolkata, India
{sushmita_t,pmaji}@isical.ac.in

Abstract. Selection of reliable genes from a huge gene expression data containing high intergene correlation is essential to carry out a diagnostic test and successful treatment. In this regard, a rough set based gene selection algorithm is reported, which selects a set of genes by maximizing the relevance and significance of the selected genes. A gene ontology-based similarity measure is proposed to analyze the functional diversity of the selected genes. It also helps to analyze the effectiveness of different gene selection methods. The performance of the rough set based gene selection algorithm, along with a comparison with other gene selection methods, is studied using the predictive accuracy of K-nearest neighbor rule and support vector machine on two cancer and one arthritis microarray data sets. An important finding is that the rough set based gene selection algorithm selects more functionally diverse set of genes than the existing algorithms.

Keywords: Rough sets, Gene selection, Functional diversity, Gene ontology.

1 Introduction

A microarray gene expression data set can be represented by an expression table, where each row corresponds to one particular gene, each column to a sample or time point, and each entry of the matrix is the measured expression level of a particular gene in a sample or time point, respectively [2]. However, among the large amount of genes, only a small fraction is desirable in developing gene expression based diagnostic tools for delivering precise, reliable, and interpretable results. Hence, identifying a reduced set of most relevant and significant genes is the goal of gene selection. The small number of training samples and a large number of genes make it a challenging problem for discriminant analysis of microarray data.

In this background, Loennstedt and Speed [6] developed a gene selection algorithm based on the empirical bayes moderated t-statistic that ranks genes by testing whether all pairwise contrasts between different classes are zero. On the other hand, significance analysis of microarrays (SAM) method [9] assigns a score to each gene on the basis of change in gene expression relative to the standard deviation of repeated measurements. All of these methods are univariate and donot consider the effect of one gene on another gene. Hence, the results obtained by these methods contain redundant gene set and may not be able to generate functionally diverse gene set.

The main objective of the current research work is to select a set of genes, which are functionally diverse and contain least redundant information. In this regard, a rough set

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 477–484, 2011.
© Springer-Verlag Berlin Heidelberg 2011

based gene selection algorithm [7] is reported to select a set of genes from microarray gene expression data by maximizing both relevance and significance of the selected genes. It employs rough set theory to compute the relevance and significance of the genes. A method is introduced to identify the optimum values of different parameters used in the rough set based gene selection algorithm. The performance of the rough set based gene selection algorithm is compared with that of existing approaches using the predictive accuracy of K-nearest neighbor rule and support vector machine on three microarray data sets. A gene ontology-based similarity assessment has been done for the genes selected by different gene selection algorithms to understand the functional diversity of genes.

2 Rough Set Based Gene Selection Algorithm

In real data analysis such as microarray data, the data set may contain a number of insignificant genes. The presence of such irrelevant and insignificant genes may lead to a reduction in the useful information. Ideally, the selected genes should have high relevance with the classes and high significance in the gene set and thus the set enhances the predictive capability. Accordingly, a measure is required that can enhance the effectiveness of gene set. In this regard, a gene selection algorithm is developed in [7], which is based on the theory of rough sets to select the relevant and significant genes from high dimensional microarray data. Let $\mathbb{C} = \{\mathscr{A}_1, \cdots, \mathscr{A}_i, \cdots, \mathscr{A}_j, \cdots, \mathscr{A}_m\}$ denotes the set of m features or genes of a given microarray data set and $\mathbb{S}$ is the set of selected genes. Define $\hat{f}(\mathscr{A}_i, \mathbb{D})$ as the relevance of the gene $\mathscr{A}_i$ with respect to the class labels $\mathbb{D}$ while $\tilde{f}(\mathscr{A}_i, \mathscr{A}_j)$ as the significance of the gene $\mathscr{A}_j$ with respect to the gene $\mathscr{A}_i$. Hence, the total relevance of all selected genes and the total significance among the selected genes are given by

$$\mathscr{J}_{\text{relev}} = \sum_{\mathscr{A}_i \in \mathbb{S}} \hat{f}(\mathscr{A}_i, \mathbb{D}); \ \mathscr{J}_{\text{signf}} = \sum_{\mathscr{A}_i \neq \mathscr{A}_j \in \mathbb{S}} \tilde{f}(\mathscr{A}_i, \mathscr{A}_j).$$

Therefore, the problem of selecting a set $\mathbb{S}$ of relevant and significant genes from the whole set $\mathbb{C}$ of m genes is equivalent to maximize both $\mathscr{J}_{\text{relev}}$ and $\mathscr{J}_{\text{signf}}$, that is, to maximize the objective function $\mathscr{J}$, where

$$\mathscr{J} = \mathscr{J}_{\text{relev}} + \beta \mathscr{J}_{\text{signf}} = \sum_{\mathscr{A}_i \in \mathbb{S}} \hat{f}(\mathscr{A}_i, \mathbb{D}) + \beta \sum_{\mathscr{A}_i \neq \mathscr{A}_j \in \mathbb{S}} \tilde{f}(\mathscr{A}_i, \mathscr{A}_j) \tag{1}$$

where β is a weight parameter. Both the relevance and significance are calculated based on the theory of rough set. The main steps of this algorithm are presented in [7].

2.1 Generation of Equivalence Classes

To calculate relevance and significance of genes using rough set theory, the continuous expression values of a gene have to be divided into several discrete partitions to generate equivalence classes. However, the inherent error that exists in conventional discretization processes is of major concern in the computation of relevance and significance of continuous valued genes. To address this problem, a fuzzy set based discretization method is used in [7] to generate equivalence classes required to compute both relevance and significance of genes using rough set theory. Given a finite set $\mathbb{U}$, $\mathbb{C}$ is a fuzzy condition attribute set in $\mathbb{U}$, which generates a fuzzy equivalence partition on $\mathbb{U}$. If c denotes the number of fuzzy equivalence classes generated by the fuzzy equivalence relation and n is the number of objects in $\mathbb{U}$, then a matrix $\mathbb{M}_{\mathbb{C}}$ of size $(c \times n)$ containing elements $\{\mu_{ij}^{\mathbb{C}}\}$ can be generated $\mathbb{M}_{\mathbb{C}} = [\mu_{ij}^{\mathbb{C}}]$, which is denoted by

$$\mathbb{M}_{\mathbb{C}} = \begin{pmatrix} \mu_{11}^{\mathbb{C}} & \mu_{12}^{\mathbb{C}} & \cdots & \mu_{1n}^{\mathbb{C}} \\ \mu_{21}^{\mathbb{C}} & \mu_{22}^{\mathbb{C}} & \cdots & \mu_{2n}^{\mathbb{C}} \\ \cdots & \cdots & \cdots & \cdots \\ \mu_{c1}^{\mathbb{C}} & \mu_{c2}^{\mathbb{C}} & \cdots & \mu_{cn}^{\mathbb{C}} \end{pmatrix} \tag{2}$$

where $\mu_{ij}^{\mathbb{C}} \in [0,1]$ represents the membership of object x_j in the ith fuzzy equivalence partition or class F_i. The above axioms should hold for every fuzzy equivalence partition, which correspond to the requirement that an equivalence class is nonempty. In the rough set based gene selection method, the π function in one dimensional form is used to assign membership values to different fuzzy equivalence classes for the input genes. A fuzzy set with membership function $\pi(x;\bar{c},\sigma)$ represents a set of points clustered around $\bar{c}$, where

$$\pi(x;\bar{c},\sigma) = \begin{cases} 2(1-\frac{||x-\bar{c}||}{\sigma})^2 & \text{for } \frac{\sigma}{2} \leq ||x-\bar{c}|| \leq \sigma \\ 1-2(\frac{||x-\bar{c}||}{\sigma})^2 & \text{for } 0 \leq ||x-\bar{c}|| \leq \frac{\sigma}{2} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where $\sigma > 0$ is the radius of the π function with $\bar{c}$ as the central point and $||\cdot||$ denotes the Euclidean norm. The $(c \times n)$ matrix $\mathbb{M}_{\mathscr{A}_i}$, corresponding to the ith gene $\mathscr{A}_i$, can be calculated from the c-fuzzy equivalence classes of the objects $x = \{x_1, \cdots, x_j, \cdots, x_n\}$, where

$$\mu_{kj}^{\mathscr{A}_i} = \frac{\pi(x_j;\bar{c}_k,\sigma_k)}{\sum_{l=1}^{c}\pi(x_j;\bar{c}_l,\sigma_l)}. \tag{4}$$

In effect, each position $\mu_{kj}^{\mathscr{A}_i}$ of the matrix $\mathbb{M}_{\mathscr{A}_i}$ must satisfy the following conditions:

$$\mu_{kj}^{\mathscr{A}_i} \in [0,1]; \ \sum_{k=1}^{c}\mu_{kj}^{\mathscr{A}_i} = 1, \forall j \text{ and for any value of } k, \text{ if}$$

$$s = \arg\max_j\{\mu_{kj}^{\mathscr{A}_i}\}, \text{ then } \max_j\{\mu_{kj}^{\mathscr{A}_i}\} = \max_l\{\mu_{ls}^{\mathscr{A}_i}\} > 0.$$

After the generation of the matrix $\mathbb{M}_{\mathscr{A}_i}$ corresponding to the gene $\mathscr{A}_i$, the object x_j is assigned to one of the c equivalence classes based on the maximum value of memberships of the object in different equivalence classes that follows next:

$$x_j \in F_p, \qquad \text{where } p = \arg\max_k\{\mu_{kj}^{\mathscr{A}_i}\}.$$

Each input real valued gene in quantitative form can be assigned to different fuzzy equivalence classes in terms of membership values using the π fuzzy set with appropriate $\bar{c}$ and σ. In the rough set based gene selection algorithm, three fuzzy equivalence classes (c=3), namely, low, medium, and high are considered. These three equivalence classes correspond to under-expression, base-line, and over-expression of continuous valued genes, respectively and also hold the following relations:

$$\bar{c}_1 = \bar{c}_{\text{low}}(\mathscr{A}_i);\ \bar{c}_2 = \bar{c}_{\text{medium}}(\mathscr{A}_i);\ \bar{c}_3 = \bar{c}_{\text{high}}(\mathscr{A}_i);$$
$$\sigma_1 = \sigma_{\text{low}}(\mathscr{A}_i);\ \sigma_2 = \sigma_{\text{medium}}(\mathscr{A}_i);\ \sigma_3 = \sigma_{\text{high}}(\mathscr{A}_i).$$

The parameters $\bar{c}$ and σ of each π fuzzy set are computed according to the following procedure [8]. Let $\bar{m}_i$ be the mean of the objects $x = \{x_1, \cdots, x_j, \cdots, x_n\}$ along the ith gene $\mathscr{A}_i$. Then $\bar{m}_{i_l}$ and $\bar{m}_{i_h}$ are defined as the mean along the ith gene of the objects having co-ordinate values in the range $[\mathscr{A}_{i_{\min}}, \bar{m}_i)$ and $(\bar{m}_i, \mathscr{A}_{i_{\max}}]$, respectively, where $\mathscr{A}_{i_{\max}}$ and $\mathscr{A}_{i_{\min}}$ denote the upper and lower bounds of the dynamic range of gene $\mathscr{A}_i$ for the training set. For three fuzzy sets low, medium, and high, the centers and corresponding radii are computed as follows:

$$\bar{c}_{\text{low}}(\mathscr{A}_i) = \bar{m}_{i_l};\ \bar{c}_{\text{medium}}(\mathscr{A}_i) = \bar{m}_i;\ \bar{c}_{\text{high}}(\mathscr{A}_i) = \bar{m}_{i_h};\ \sigma_{\text{medium}}(\mathscr{A}_i) = \eta \times \frac{\text{A}}{\text{B}};$$
$$\sigma_{\text{low}}(\mathscr{A}_i) = 2[\bar{c}_{\text{medium}}(\mathscr{A}_i) - \bar{c}_{\text{low}}(\mathscr{A}_i)];\ \sigma_{\text{high}}(\mathscr{A}_i) = 2[\bar{c}_{\text{high}}(\mathscr{A}_i) - \bar{c}_{\text{medium}}(\mathscr{A}_i)];$$

$$\text{where}\ \ \text{A} = \{\sigma_{\text{low}}(\mathscr{A}_i)(\mathscr{A}_{i_{\max}} - c_{\text{medium}}(\mathscr{A}_i)) +$$
$$\sigma_{\text{high}}(\mathscr{A}_i)(c_{\text{medium}}(\mathscr{A}_i) - \mathscr{A}_{i_{\min}})\}; \text{B} = \{\mathscr{A}_{i_{\max}} - \mathscr{A}_{i_{\min}}\}$$

where η is a multiplicative parameter controlling the extent of the overlapping between three fuzzy equivalence classes low and medium or medium and high.

3 Experimental Results and Discussions

The performance of the rough set based gene selection method (MRMS) [7] is extensively studied and compared with that of some existing algorithms, namely, Hall's combinatorial feature selection algorithm (CFS) [5], Ebayes, Ensemble, partial least squares cross-validation method (PLSCV) [5], random forest feature selection (RFMDA) [1], and SAM [9]. To analyze the performance of different gene selection methods, the experimentation is done on two cancer (Breast, and Leukemia) and one arthritis (RAOA) microarray data sets. For each data set, fifty top-ranked genes are selected for analysis. The performance of different gene selection methods is evaluated by using the classification accuracy of K-nearest neighbor (K-NN) rule and support vector machine (SVM). To compute the prediction accuracy of both SVM and K-NN rule, the leave-one-out cross-validation (LOOCV) is performed on each microarray data set. The biological evaluation of these methods is also conducted using G-Sesame [3], which is developed using the concept of gene ontology.

3.1 Optimum Values of Parameters

The weight parameter β in (1) plays an important role to reduce redundancy in between genes present in microarray data. In effect, it has a direct influence on the performance of the MRMS algorithm. Also, the MRMS algorithm depends on multiplicative parameter η. Let $S=\{\beta,\eta\}$ be the set of parameters and $S^*=\{\beta^*,\eta^*\}$ is the set of optimal parameters. To find out the optimum set S^*, containing optimum values of β^* and η^*, the support vector machine [10] and K-nearest neighbor (K-NN) [4] are used.

Support Vector Machine. The support vector machine (SVM) [10] is a margin classifier that draws an optimal hyperplane in the feature vector space; this defines a boundary that maximizes the margin between data samples in different classes, therefore leading to good generalization properties. A key factor in the SVM is to use kernels to construct nonlinear decision boundary. In the present work, linear kernels are used.

K-Nearest Neighbor Rule. The K-nearest neighbor (K-NN) rule [4] is used for evaluating the effectiveness of the reduced gene set for classification. It classifies samples based on closest training samples in the feature space. A sample is classified by a majority vote of its K-neighbors, with the sample being assigned to the class most common amongst its K-nearest neighbors. The value of K, chosen for the K-NN, is the square root of the number of samples in training set.

For three microarray data sets, the value of β is varied from 0.0 to 1.0, while the value of η is varied from 0.5 to 2.0. The optimum values of β^* and $\eta^*$for each microarray data set are obtained using the following relation:

$$S^* = \arg\max_{S}\{\text{Classification accuracy of D}\};\ \text{where D is either SVM or KNN.} \quad (5)$$

The optimum sets S^* containing optimum values of β and η obtained using (5) are $\{0.3,0.8\}, \{0.1,1.5\}$, and $\{0.6,0.8\}$ for Breast, Leukemia, and RAOA data sets, respectively using the SVM, while $\{0.6,1.2\}, \{0.1,1.7\}$, and $\{0.7,0.9\}$ for Breast, Leukemia, and RAOA data sets, respectively using the KNN.

3.2 Comparative Performance Analysis of Different Algorithms

The best results of different algorithms on three microarray data sets are presented in Table 1. The classification accuracy of both SVM and K-NN of the MRMS is better compared to other gene selection algorithms in all cases. The best performance of the MRMS algorithm, in terms of accuracy of both SVM and K-NN, is achieved due to the maximization of relevance and significance of genes simultaneously.

3.3 Gene Ontology Based Analysis to Study Independency of Selected Genes

The web based tool G-SESAME [3] is used to measure functional similarity between a pair of genes obtained by various gene selection algorithms. This tool quantifies functional similarity between a pair of genes based on gene annotation information from heterogeneous data sources. The value of similarity ranges from zero to one, nearer

Table 1. Performance of Different Gene Selection Algorithms

Data Sets	Methods/ Algorithms	SVM		K-NN	
		Accuracy	Number of genes	Accuracy	Number of genes
Breast	CFS	97.96	28	97.96	20
	EBayes	93.88	4	93.88	24
	ENSEMBLE	95.92	6	91.84	8
	PLSCV	95.92	6	95.92	9
	RFMDA	95.92	5	93.88	15
	SAM	93.88	7	93.88	18
	MRMS	100.00	6	100.00	6
Leukemia	CFS	100.00	7	98.61	21
	EBayes	97.22	36	97.22	34
	ENSEMBLE	98.61	15	97.22	14
	PLSCV	100.00	47	97.22	38
	RFMDA	97.22	7	97.22	11
	SAM	98.61	43	97.22	45
	MRMS	100.00	4	100.00	3
RAOA	CFS	100.00	7	93.33	2
	EBayes	93.33	1	90.00	1
	ENSEMBLE	100.00	7	96.67	48
	PLSCV	100.00	7	96.67	48
	RFMDA	100.00	7	96.67	48
	SAM	100.00	7	96.67	48
	MRMS	100.00	4	100.00	8

the value to one higher the functional similarity between the pair of genes. Given two genes G1 and G2 and their annotated GO terms GO_1=$\{go_{11}$,go_{12},...,$go_{1m}\}$ and GO_2= $\{go_{21}$,go_{22},...,$go_{2n}\}$, their functional similarity is define as:

$$Sim(G1,G2)=\frac{\sum_{1=1}^{m} Sim(go_{1i},GO_2)+\sum_{1=j}^{n} Sim(go_{2j},GO_1)}{m+n} \tag{6}$$

In the present work, the G-SESAME is used to quantify functional diversity present in between selected genes. The numbers of gene pairs obtained by G-SESAME tool for a dataset in all methods and in all ontologies are not uniform. To make the comparison between all methods, a measure called degree of functional diversity (*DoFD*), is proposed next. For a given distribution of similarity (C), the *DoFD* for each method can be calculated by

$$DoFD(C)=\sum_{i=1}^{5} w_i n_i \tag{7}$$

where w_i=$\{0.50,0.25,0.125,0.0625,0.0625\}$ is weight parameter and n_i is the fraction of each category of similarity distribution table, which is obtained by taking ratio of frequency and gene pairs number. The frequency is calculated by counting the number of gene pairs which occur in a particular category. Here, five categories are generated. Gene pairs having semantic similarity in range of 0.0 to 0.5, signifies that the gene pairs are functionally very diverse. Higher the value of *DoFD* means higher functional diversity. This tool is applied on gene sets selected by CFS, Ebayes, Ensemble, PLSCV, RFMDA, SAM, and MRMS methods on three datasets. The functional diversity is calculated for three ontologies, namely, biological processes (BPs), molecular functions

Table 2. Degree of Functional Diversity and Distribution of Best Selected Genes

Microarray Data Sets	Different Ontologies	Methods/ Algorithms	*DoFD*	Similarity Distribution									
				0.0-0.2		0.2-0.4		0.4-0.6		0.6-0.8		0.8-1.0	
				Freq.	Frac.	Freq.	Frac.	Freq.	Frac.	Freq.	Frac.	Freq.	Frac.
Breast	MF	CFS(483)	0.672	254	0.526	69	0.143	130	0.269	21	0.043	9	0.019
		Ebayes(1015)	0.709	596	0.587	117	0.115	220	0.217	42	0.041	40	0.039
		Ensemble(924)	0.641	460	0.498	121	0.131	229	0.248	69	0.075	45	0.049
		PLSCV(648)	0.662	359	0.554	48	0.074	128	0.198	75	0.116	38	0.059
		RFMDA(1003)	0.635	502	0.500	111	0.111	247	0.246	99	0.099	44	0.044
		SAM(885)	0.755	580	0.655	78	0.088	164	0.185	24	0.027	39	0.044
		MRMS(SVM)(360)	0.964	337	0.940	15	0.040	4	0.010	4	0.010	0	0.000
		MRMS(K-NN)(220)	**0.980**	214	0.970	5	0.020	1	0.000	0	0.000	0	0.000
	BP	CFS(525)	0.833	365	0.695	131	0.250	27	0.051	2	0.004	0	0.000
		Ebayes(1060)	0.865	798	0.753	218	0.206	36	0.034	6	0.006	2	0.002
		Ensemble(935)	0.864	708	0.757	178	0.190	40	0.043	6	0.006	3	0.003
		PLSCV(682)	0.815	464	0.680	167	0.245	16	0.023	11	0.016	24	0.035
		RFMDA(1053)	0.836	739	0.702	258	0.245	41	0.039	12	0.011	3	0.003
		SAM(969)	0.874	749	0.773	177	0.183	31	0.032	8	0.008	4	0.004
		MRMS(SVM)(395)	0.963	369	0.930	23	0.060	2	0.010	1	0.000	0	0.000
		MRMS(K-NN)(220)	**0.980**	214	0.970	5	0.020	1	0.000	0	0.000	0	0.000
	CC	CFS(518)	0.551	180	0.347	123	0.237	135	0.261	64	0.124	16	0.031
		Ebayes(1036)	0.596	428	0.413	224	0.216	234	0.226	122	0.118	28	0.027
		Ensemble(924)	0.642	439	0.475	202	0.219	139	0.150	104	0.113	40	0.043
		PLSCV(648)	0.684	356	0.549	115	0.177	59	0.091	60	0.093	58	0.090
		RFMDA(961)	0.670	522	0.543	113	0.118	194	0.202	106	0.110	26	0.027
		SAM(924)	0.632	441	0.477	165	0.179	167	0.181	120	0.130	31	0.034
		MRMS(SVM)(360)	0.940	326	0.910	14	0.040	12	0.030	5	0.010	3	0.010
		MRMS(K-NN)(220)	**0.973**	212	0.960	5	0.020	3	0.010	0	0.000	0	0.000
Leukemia	MF	CFS(435)	0.447	100	0.230	91	0.209	148	0.340	73	0.168	23	0.053
		Ebayes(1036)	0.686	603	0.582	88	0.085	163	0.157	140	0.135	42	0.041
		Ensemble(891)	0.663	497	0.558	81	0.091	116	0.130	150	0.168	47	0.053
		PLSCV(806)	0.550	341	0.423	53	0.066	195	0.242	103	0.128	114	0.141
		RFMDA(1036)	0.621	496	0.479	140	0.135	216	0.208	148	0.143	36	0.035
		SAM(999)	0.646	540	0.541	71	0.071	172	0.172	164	0.164	52	0.052
		MRMS(SVM)(703)	0.840	549	0.780	50	0.070	38	0.050	53	0.080	13	0.020
		MRMS(K-NN)(616)	**0.881**	510	0.830	34	0.060	37	0.060	27	0.040	8	0.010
	BP	CFS(432)	0.869	325	0.752	95	0.220	10	0.023	1	0.002	1	0.002
		Ebayes(990)	0.894	799	0.807	158	0.160	26	0.026	6	0.006	1	0.001
		Ensemble(868)	0.897	703	0.810	140	0.161	17	0.020	7	0.008	1	0.001
		PLSCV(806)	0.863	597	0.741	189	0.234	14	0.017	5	0.006	1	0.001
		RFMDA(1003)	0.907	832	0.830	141	0.141	24	0.024	5	0.005	1	0.001
		SAM(957)	0.894	772	0.807	155	0.162	22	0.023	7	0.007	1	0.001
		MRMS(SVM)(675)	0.935	603	0.890	55	0.080	13	0.020	3	0.000	1	0.000
		MRMS(K-NN)(646)	**0.950**	589	0.910	43	0.070	10	0.020	3	0.000	1	0.000
	CC	CFS(434)	0.501	96	0.221	174	0.401	112	0.258	47	0.108	5	0.012
		Ebayes(1036)	0.653	510	0.492	204	0.197	191	0.184	106	0.102	25	0.024
		Ensemble(918)	0.614	389	0.424	221	0.241	203	0.221	87	0.095	18	0.020
		PLSCV(783)	0.666	408	0.521	135	0.172	127	0.162	100	0.128	13	0.017
		RFMDA(1045)	0.613	443	0.424	251	0.240	227	0.217	104	0.100	20	0.019
		SAM(1007)	0.611	426	0.423	245	0.243	198	0.197	110	0.109	28	0.028
		MRMS(SVM)(585)	0.864	480	0.820	25	0.040	38	0.060	25	0.040	17	0.030
		MRMS(K-NN)(585)	**0.878**	480	0.820	45	0.080	23	0.040	24	0.040	13	0.020

(MFs), or cellular components (CCs). The *DoFD* values measured by (7) and the similarity distribution over all gene pairs are reported in Table 2. The values in bracket mentioned after the name of each method depict the obtained gene pair numbers which have semantic similarity. All the top 50 genes selected by different gene selection algorithms are used in this analysis. In case of the MRMS algorithm, SVM and K-NN classifiers generate different sets of optimum values of β and η parameters. So, in gene ontology based analysis, results are presented for optimum values of β and η obtained by both SVM and K-NN. From Table 2, it can be observed that the similarity distribution of most gene pairs in the gene sets are located at the lower end for all datasets and in three ontologies. In all ontologies and datasets, the *DoFD* values of genes of the MRMS method are higher than that obtained other gene selection algorithms.

4 Conclusion

In this paper, an application of rough set has been demonstrated to select functionally diverse genes from microarray data. For two cancer and one arthritis microarray data sets, significantly better results are found for the rough set based method compared to existing gene selection methods. The rough set based algorithm generates more functionally diverse sets of genes, which also provide high classification accuracy. All the results reported in this paper demonstrate the feasibility and effectiveness of the rough set based method.

Acknowledgement. The work was done when one of the authors, S. Paul, was a Senior Research Fellow of Council of Scientific and Industrial Research, Government of India.

References

1. Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)
2. Domany, E.: Cluster Analysis of Gene Expression Data. Journal of Statistical Physics 110(3-6), 1117–1139 (2003)
3. Du, Z., Li, L., Chen, C.F., Yu, P.S., Wang, J.Z.: G-sesame: Web tools for go-term-based gene similarity analysis and knowledge discovery. Nucleic Acids Research 37, W345–W349 (2009)
4. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification and Scene Analysis. John Wiley and Sons, New York (1999)
5. Hall, M.: Correlation-Based Feature Selection for Discrete and Numeric Class Machine Learning. In: Proceedings of the Seventeenth International Conference on Machine Learning, pp. 359–366 (2000)
6. Loennstedt, I., Speed, T.P.: Replicated microarray data. Statistica Sinica 12, 31–46 (2002)
7. Maji, P., Paul, S.: Rough set based maximum relevance-maximum significance criterion and gene selection from microarray data. International Journal of Approximate Reasoning 52(3), 408–426 (2011)
8. Pal, S.K., Mitra, S.: Neuro-Fuzzy Pattern Recognition: Methods in Soft Computing. Wiley, New York (1999)
9. Tusher, V., Tibshirani, R., Chu, G.: Significance analysis of microarrays applied to the ionizing radiation response. Proceedings of the National Academy of Sciences 98, 5116–5121 (2001)
10. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1995)

Quality Evaluation Measures of Pixel - Level Image Fusion Using Fuzzy Logic

Srinivasa Rao Dammavalam[1], Seetha Maddala[2], and M.H.M. Krishna Prasad [3]

[1] Department of Information Technology, VNRVJIET, Hyderabad, India
[2] Department of CSE, GNITS, Hyderabad, India
[3] Department of CSE, JNTU College of Engineering, Vizianagaram, India
{dammavalam2,krishnaprasad.mhm}@gmail.com,
smaddala2000@yahoo.com,

Abstract. Image fusion is a technique to combine the registered images to increase the spatial resolution of acquired low detail multi-sensor images and preserving their spectral information. In fusing panchromatic and multispectral images, the objective is to reduce uncertainty and minimize redundancy in the output while maximizing relevant information. Different fusion methods provide different results for different applications, medical imaging, automatic target guidance system, remote sensing, machine vision, automatic change detection, and biometrics. In this paper, we utilize a fuzzy logic approach to fuse images from different sensors, in order to enhance visualization. The work here further explores the comparison between image fusion using wavelet transform and fuzzy logic approach along with performance/quality evaluation measures like image quality index, entropy, mutual information measure, root mean square error, peak signal to noise ratio, fusion factor, fusion symmetry and fusion index. Experimental results prove that the use of the proposed method can efficiently preserve the spectral information while improving the spatial resolution of the remote sensing images.

Keywords: image fusion, panchromatic, multispectral, wavelet transform, fuzzy logic, mutual information measure, image quality index, fusion factor, fusion symmetry, fusion index, entropy.

1 Introduction

Image fusion is a process of integrating information obtained from various sensors and intelligent systems. It provides a single image containing complete information. The concept of Image fusion has been used in wide variety of applications like medical imaging, remote sensing, navigation aid, machine vision, automatic change detection, biometrics etc. Yi Zheng et al. [1] proposed a method, Multisensor image fusion for surveillance systems in which fuzzy logic approach utilized to fuse images from different sensors, in order to enhance visualization for surveillance. In [2] Xu-Hong Yang et. al. proposed, urban remote image fusion using fuzzy rules to refine the

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 485–493, 2011.
© Springer-Verlag Berlin Heidelberg 2011

resolution of urban multi-spectral images using the corresponding high-resolution panchromatic images**.** After the decomposition of two input images by wavelet transform three texture features are extracted and then a fuzzy fusion rule is used to merge wavelet coefficients from the two images according to the extracted features. Zhu Mengyu et al. [3] present a image fusion algorithm based on fuzzy logic and wavelet, aimed at the visible and infrared image fusion and address an algorithm based on the discrete wavelet transform and fuzzy logic. In [3] the technique created two fuzzy relations, and estimated the importance of every wavelet coefficient with fuzzy reasoning. In [4] Rahul Ranjan et al. give a Iterative Fuzzy and Neuro Fuzzy approach for fusing medical images and remote sensing images and found that the technique very useful in medical imaging and other areas, where quality of image is more important than the real time application. In [5] a new method is proposed for Pixel-Level Multisensor image fusion based on Fuzzy Logic in which the membership function and fuzzy rules of the new algorithm is defined using the Fuzzy Inference System. Yang-Ping Wang et al. [6] proposed, a fuzzy radial basis function neural networks is used to perform auto-adaptive image fusion and in experiment multimodal medical image fusion based on gradient pyramid is performed for comparison. In [7] a novel method is proposed using combine framework of wavelet transform and fuzzy logic and it provides novel tradeoff solution between the spectral and spatial fidelity and preserves more detail spectral and spatial information. Bushra et al. [8] proposed a method, Pixel & Feature Level Multi-Resolution Image Fusion based on Fuzzy logic in which images are first segmented into regions with fuzzy clustering and are then fed into a fusion system, based on fuzzy if-then rules.

2 Image Fusion Using Wavelet Transform

Wavelet Transform is a type of signal representation that can give the frequency content of the signal at a particular instant of time. Wavelet analysis has advantages over traditional Fourier methods in analyzing physical situations where the signal contains discontinuities and sharp spikes. Li et al. [9] proposed a multi-sensor image fusion using the wavelet transform method that the wavelet transforms of the input images are appropriately combined, and the new image is obtained by taking the inverse wavelet transform of the fused wavelet coefficients. An area-based maximum selection rule and a consistency verification step are used for feature selection. In [10] Wavelet transforms provide a framework in which a signal is decomposed, with each level corresponding to a coarser resolution, or lower frequency band. There are two main groups of transforms, continuous and discrete. Discrete transforms are more commonly used and can be subdivided in various categories. Susmitha et al. [11] Proposed a novel architecture for wavelet based fusion of images from different sources using multiresolution wavelet transforms which applies pixel based maximum selection rule to low frequency approximations and filter mask based fusion to high frequency details of wavelet decomposition. The key feature of hybrid architecture is the combination of advantages of pixel and region based fusion in a single image which can help the development of sophisticated algorithms enhancing the edges and structural details. Krista et al. [12] give a comprehensive literature survey of wavelet transform theory and an overview of image fusion techniques, and the results from a number of wavelet-based image fusion schemes are compared.

3 Image Fusion Based on Fuzzy Logic

It is important to know that the set of images used in this algorithm are registered images. Through registration we find correspondence between images and ensured that spatial correspondence established, fusion makes sense [8].

3.1 Fuzzy Logic in Image Processing

Fuzzy logic has three main stages. Image fuzzification, modification of membership values, image defuzzification. The coding of image data (fuzzification) and decoding of the results (defuzzification) are steps that make possible to process images with fuzzy techniques. The main power of fuzzy image processing is in the middle step (modification of membership values). After the image data are transformed from gray-level plane to the membership plane (fuzzification), appropriate fuzzy techniques modify the membership values. It can be a fuzzy clustering, a fuzzy rule-based, fuzzy integration approach and so on [13].

3.2 Steps Involved in Fuzzy Image Processing

The original image in the gray level plane is subjected to fuzzification and the modification of membership functions is carried out in the membership plane. The result is the output image obtained after the defuzzification process.

The first step fuzzification is done as follows in both the source images.

- Both the images are compared with each other.

The second step Modification of memberships is done as follows:

- If $f_a(x) = f_b(x)$ then $F_f(x) = 0$
- If $f_a(x) > f_b(x)$ then $F_f(x) = 1$
- If $f_a(x) < f_b(x)$ then $F_f(x) = -1$

The third step defuzzification is done as follows:

- If $F_f(x) = 0$ then $F_f(x) = f_a(x)$ or $f_b(x)$
- If $F_f(x) = 1$ then $F_f(x) = f_a(x)$
- If $F_f(x) = -1$ then $F_f(x) = f_b(x)$

Membership functions and rules considered in the fuzzy system

1. if (input1 is mf2) and (input2 is mf3) then (output1 is mf2)
2. if (input1 is mf1) and (input2 is mf2 then (output1 is mf3)
3. if (input1 is mf3) and (input2 is mf2) then (output1 is mf2)

Algorithm steps for pixel level image fusion using Fuzzy Logic approach [14].

- Read first image in variable I1 and find its size (rows: rl, columns: c1).
- Read second image in variable I2 and find its size (rows:r2, columns: c2).
- Variables I1 2 and I2 are images in matrix form where each pixel value is in the range from 0-255. Use Gray Colormap.
- Compare rows and columns of both input images. If the two images are not of the same size, select the portion, which are of same size.
- Convert the images in column form which has C= rl*cl entries.
- Make a fis (Fuzzy) file, which has two input images.
- Decide number and type of membership functions for both the input images by tuning the membership functions. Input images in antecedent are resolved to a degree of membership ranging 0 to 255.
- Make rules for input images, which resolve the two antecedents to a single number from 0 to 255.
- For num=l to C in steps of one, apply fuzzification using the rules developed above on the corresponding pixel values of the input images which gives a fuzzy set represented by a membership function and results in output image in column format.
- Convert the column form to matrix form and display the fused image.

4 Evaluation Measures

Evaluation measures are used to evaluate the quality of the fused image. The fused images are evaluated, taking the following parameters into consideration.

4.1 Image Quality Index

IQI measures the similarity between two images (I1 & I2) and its value ranges from -1 to 1. IQI is equal to 1 if both images are identical. IQI measure is given by [15]

$$IQI = \frac{m_{ab}\ 2xy\ 2m_a m_b}{m_a m_b x^2 + y^2 m_a^2 + m_b^2} \tag{1}$$

Where x and y denote the mean values of images I1 and I2 and m_a^2, m_b^2 and m_{ab} denotes the variance of I1, I2 and covariance of I1 and I2.

4.2 Fusion Factor

Given two images A and B, and their fused image F, the Fusion factor (FF) is illustrated as [16]:

$$FF = I_{AF} + I_{BF} \tag{2}$$

Where I_{AF} and I_{BF} are the MIM values between input images and fused image. A higher value of *FF* indicates that fused image contains moderately good amount of information present in both the images. However, a high value of *FF* does not imply that the information from both images is symmetrically fused.

4.3 Fusion Symmetry

Fusion symmetry (FS) is an indication of the degree of symmetry in the information content from both the images.

$$FS = abs\left(\frac{I_{AF}}{I_{AF} + I_{BF}} - 0.5\right) \tag{3}$$

The quality of fusion technique depends on the degree of Fusion symmetry. Since FS is the symmetry factor, when the sensors are of good quality, FS should be as low as possible so that the fused image derives features from both input images. If any of the sensors is of low quality then it is better to maximize FS than minimizing it.

4.4 Fusion Index

This study proposes a parameter called Fusion index from the factors Fusion symmetry and Fusion factor. The fusion index (FI) is defined as

$$FI = I_{AF} / I_{BF} \tag{4}$$

Where I_{AF} is the mutual information index between multispectral image and fused image and I_{BF} is the mutual information index between panchromatic image and fused image. The quality of fusion technique depends on the degree of fusion index.

4.5 Mutual Information Measure

Mutual information measure furnishes the amount of information of one image in another. This gives the guidelines for selecting the best fusion method. Given two images *M (i, j) and N (i, j)* and MIM between them is defined as:

$$I_{MN} = \sum_{x,y} P_{MN}(x, y) \log \frac{P_{MN}(x, y)}{P_M(x) P_N(y)} \tag{5}$$

Where, P_M (x) and P_N (y) are the probability density functions in the individual images, and $P_{MN}(x, y)$ is joint probability density function.

4.6 Root Mean Square Error

The root mean square error (RMSE) measures the amount of change per pixel due to the processing. The RMSE between a reference image R and the fused image F is given by

$$RMSE = \sqrt{\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(R(i,j) - F(i,j)\right)} \tag{6}$$

4.7 Peak Signal to Noise Ratio

Peak signal to noise ratio (PSNR) can be calculated by using the formula

$$PSNR = 20\log_{10}\left[\frac{L^2}{MSE}\right] \tag{7}$$

Where MSE is the mean square error and L is the number of gray levels in the image.

4.8 Entropy

Entropy E, a scalar value representing the entropy of grayscale image. Entropy is a statistical measure of randomness that can be used to characterize the texture of the input image. Entropy is defined as

$$E = -sum(p * \log_2(p)) \tag{8}$$

Where p contains the histogram counts returned from imhist.

5 The Experiment and Analysis

In this section, we fused a panchromatic image and multispectral image using our algorithm. Case 1, Panchromatic and Multispectral images of the Hyderabad city, AP, INDIA are acquired from the IRS 1D LISS III sensor at 05:40:44, Case 2 and Case 3 images are acquired from http://imagefusion.org [17].

The proposed algorithm has been implemented using Matlab 7.0. It can be seen from the above table and the image results that the fuzzy logic approach are having much better results when compared with the conventional technique. Table 1 shows that fuzzy based fused approach has shown comparatively better IQI, MIM and Entropy through preserving more spectral information. Considerable differences are obtained through fuzzy logic with lower RMSE and higher values of Fusion Factor and PSNR. So therefore it is concluded from experimental results that fuzzy logic based image fusion schemes perform better than conventional wavelet transform.

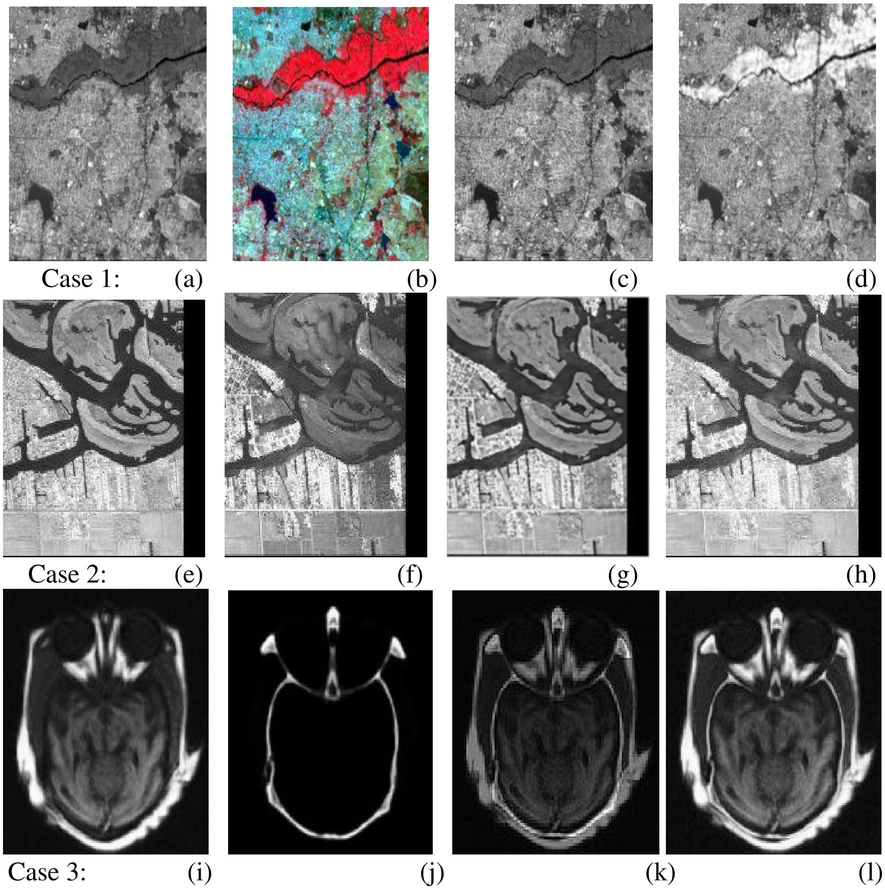


Fig. 1. Some example images (a), (b), (e), (f), (i) and (j): original input images; (c), (g) and (k): fused images by wavelet transform and (d), (h) and (l): fused images by fuzzy logic

Table 1. The evaluation measures of image fusion based on wavelet transform and fuzzy logic

Method	IQI	FF	FS	FI	MIM	RMSE	PSNR	Entropy
Wavelet Transform								
(Case 1)	0.9473	3.8629	0.0429	1.1879	1.7656	63.5529	11.3425	7.3828
(Case 2)	0.8650	3.8832	0.0118	0.9538	1.9875	19.1999	22.4648	7.2339
(Case 3)	0.5579	2.6841	0.2731	3.4074	2.0751	39.5475	16.1884	5.9807
Fuzzy Logic								
(Case 1)	0.9689	5.5687	0.2752	3.4475	4.3166	52.5301	13.7226	7.3445
(Case 2)	0.9955	8.8407	0.0598	1.2719	3.8914	17.8385	23.1036	7.2577
(Case 3)	0.9896	4.7589	0.4023	9.2320	4.2938	25.4703	20.0101	6.7300

6 Conclusions

In this paper, a pixel level image fusion using fuzzy logic approach for remote sensing and medical imaging has been presented. All the results obtained and discussed by this method are same scene. In order to evaluate the results and compare the methods, the assessment criteria, evaluation measures are employed. The experimental results clearly show that the introduction of the proposed image fusion using fuzzy logic gives a considerable improvement on the performance of the fusion system. It is hoped that the technique can be further extended to all types of images and to integrate valid evaluation metric of image fusion schemes. And automatic determinations of the percentage of overlapping among fuzzy sets and membership functions are also worthy of research.

Acknowledgments. This paper is partially supported by the All India Council for Technical Education, New Delhi, India under Research Promotion Scheme (RPS) Grant No. 8023/RID/RPS-80/2010-11.

References

1. Yi, Z., Ping, Z.: Multisensor Image Fusion Using Fuzzy Logic for Surveillance Systems. In: IEEE Seventh International Conference on Fuzzy Systems and Discovery, Shanghai, pp. 588–592 (2010)
2. Yang, X.H., Huang, F.Z., Liu, G.: Urban Remote Image Fusion Using Fuzzy Rules. In: IEEE Proceedings of the Eighth International Conference on Machine Learning and Cybernetics, Baoding, pp. 101–109 (2009)
3. Mengyu, Z., Yuliang, Y.: A New image Fusion Algorithm Based on Fuzzy Logic. In: IEEE International Conference on Intelligent Computation Technology and Automation, Changsha, pp. 83–86 (2008)
4. Ranjan, R., Singh, H., Meitzler, T., Gerhart, G.R.: Iterative Image Fusion technique using Fuzzy and Neuro fuzzy Logic and Applications. In: IEEE Fuzzy Information Processing Society, Detroit, USA, pp. 706–710 (2005)
5. Zhao, L., Xu, B., Tang, W., Chen, Z.: A Pixel-Level Multisensor Image Fusion Algorithm based on Fuzzy Logic. In: Wang, L., Jin, Y. (eds.) FSKD 2005. LNCS (LNAI), vol. 3613, pp. 717–720. Springer, Heidelberg (2005)
6. Wang, Y.P., Dang, J.W., Li, Q., Li, S.: Multimodal Medical Image fusion using Fuzzy Radial Basis function Neural Networks. In: IEEE International Conference on Wavelet Analysis and Pattern Recognition, Beijing, pp. 778–782 (2007)
7. Tanish, Z., Ishit, M., Mukesh, Z.: Novel hybrid Multispectral Image Fusion Method using Fuzzy Logic. I. J. Computer Information Systems and Industrial Management Applications, 9–103 (2010)
8. Bushra, N.K., Anwar, M.M., Haroon, I.: Pixel & Feature Level Multi-Resolution Image Fusion based on Fuzzy Logic. In: ACM Proc. of the 6th WSEAS International Conference on Wavelet Analysis & Multirate Systems, Romania, pp. 88–91 (2006)
9. Li, H., Manjunath, B.S, Mitra, S.K.: Multi-Sensor Image Fusion Using the Wavelet Transform. In: IEEE International Conference on Image Processing, Austin, pp. 51–55 (1994)

10. Kannan, K., Perumal, S.A., Arulmozhi, K.: Performance Comparision of various levels of Fusion of Multi-Focused Images using Wavelet Transform. I. J. Computer Applications (2010) ISSN 0975–8887
11. Susmitha, V., Pancham, S.: A Novel Architecture for Wavelet based Image Fusion. World Academy of Science, Engineering and Technology, 372–377 (2009)
12. Krista, A., Zhang, Y., Dare, P.: Wavelet based Image Fusion Techniques- An Introduction Review and Comparision. J. Photogrammetry & Remote Sensing, 249–263 (2007)
13. Maruthi, R., Sankarasubramanian, K.: Pixel Level Multifocus Image Fusion Based on Fuzzy Logic Approach. J. Information Technology 7(4), 168–171 (2008)
14. Thomas, M., David, B., Sohn, E.J., Kimberly, L., Darryl, B., Gulshecn, K., Harpreet, S., Samuel, E., Grmgory, S., Yelena, R., James, R.: Fuzzy Logic based Image Fusion Aerosense, Orlando (2002)
15. Mumtaz, A., Masjid, A.: Genetic Algorithms and its Applicatio to Image Fusion. In: IEEE International Conference on Emerging Technologies, Rawalpindi, pp. 6–10 (2008)
16. Seetha, M., MuraliKrishna, I.V., Deekshatulu, B.L.: Data Fusion Performance Analysis Based on Conventional and Wavelet Transform Techniques. In: IEEE Proc. Geoscience and Remote Sensing Symposium, Seoul, pp. 284–2845 (2005)
17. The Online Resource for Research in Image Fusion, `http://www.imagefusion.org`

Load Frequency Control: A Polar Fuzzy Approach

Rahul Umrao[1], D.K. Chaturvedi[2], and O.P. Malik[3]

[1] Department of Electrical Engineering Faculty of Engineering,
Dayalbagh Educational Institute, Dayalbagh, Agra, 282005, India
`rahulumrao@gmail.com`
[2] Department of Electrical Engineering Faculty of Engineering,
Dayalbagh Educational Institute, Dayalbagh, Agra, 282005, India
`dkc.foe@gmail.com`
[3] Department of Electrical and Computer Engineering,
University of Calgary, 2500 University Drive, N.W., Calgary, AB, T2N 1N4, Canada
`maliko@ucalgary.ca`

Abstract. The Performance of a Fuzzy logic controller is limited by its large number of rules and if the rules are large then computation time and requirement of memory is large. This problem is compensated by using Polar Fuzzy logic controller. So a Polar Fuzzy logic controller is proposed for the load frequency control problem. The aim of the Polar Fuzzy Controller is to restore the frequency and tie-line power in a smooth way to its nominal value in the shortest possible time if any load disturbance is applied. System performance is examined and compared with a standard Fuzzy logic controller, and conventional controllers.

1 Introduction

Today the electrical power system is very much large and made with interconnected control areas. Load Frequency Control (LFC) is an important problem in electrical power system to keep the system frequency and the inter-area tie line power as close as possible to the scheduled values for reliable electrical power with good quality [1]. By controlling the mechanical input power to the generators of plants, frequency can be maintained as scheduled value. Due to variation of load and disturbances of system, which comes any time, frequency is disturbed from its rated value. The design of power system is such that it should maintain the frequency and voltage of the network within tolerable limits [2, 3].

The main objective of the load frequency controller is to exert control of frequency and at the same time of real power exchange via out going lines [4]. Several strategies for LFC have been proposed. Although the conventional control techniques were used in majority of literature [5], several studies using novel and intelligent control techniques for LFC are also reported in the literature. These controllers have shown good results in load frequency control [6]. Application of a Polar Fuzzy controller to LFC is described in this paper.

Its performance on a single area and a two area thermal system with and without reheat unit is described, and compared with the conventional controllers like

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 494–504, 2011.
© Springer-Verlag Berlin Heidelberg 2011

proportional integral (PI), proportional integral derivative (PID), and intelligent controller like a standard Fuzzy logic controller.

2 Conventional Controllers

The PI controller is popular in the industry. The proportional gain provides stability and high frequency response. The integral term insures that the average error is driven to zero. Advantages of PI are; only two gains need to be tuned, there is no long-term error, and the method normally provides highly responsive system. The predominant weakness is that PI controllers often produce excessive overshoot to a step command [7].

Conventional PID controller is the most widely used in the industry [8-9]. The PID controller, a combination of PI and PD, can be easily adopted to correspond to different controlled plants but it cannot yield a good control performance if the controlled system is of high order and nonlinear. The PD control, as in the case of the lead compensator, improves the transient-response characteristics, improves system stability, and increases the system bandwidth, which implies fast rise time [10].

3 Fuzzy Logic Controller

After 1991 fuzzy logic application is used for industrial tools. Fuzzy controllers are preferred over conventional controllers because:

- Higher robustness.
- To develop a fuzzy controller is cheaper.
- It is easy to understand because it is expressed in natural linguistic terms.
- It is easy to learn how fuzzy controller operates and how to design.

In last two decades the fuzzy set theory is a new methodological tool for handling ambiguity and uncertainty. Load frequency control has the main goal to maintain the balance between production and consumption [11-12]. The power system is very complex in nature because of there are several variable conditions. So the fuzzy logic helps to develop robust and reliable controller for load frequency control problem [3-5, 10].

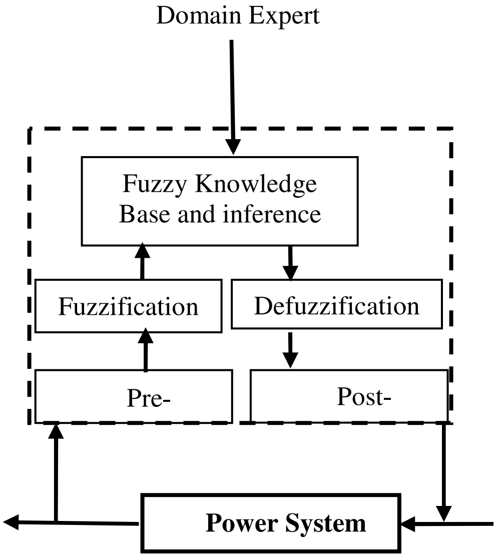


Fig. 1. Functional Module of a Simple Fuzzy Controller

Functional module of a simple fuzzy logic controller (FLC) is shown in Fig. 1.These controllers are useful for solving a wide range of control problems [13]. Five triangular, input and output both, membership functions are taken for design of FLC.

3.1 Fuzzy Rules

In FLC, five rules are used. These are:

I. If input is Z then output is LN.
II. If input is LN then output is LP.
III. If input is LP then output is LP.
IV. If input is MP then output is Z.
V. If input is MP then output is Z.

4 Polar Fuzzy Controller

The working of Polar fuzzy controller (PFC) is based on polar fuzzy sets. One important difference between the standard fuzzy sets and polar fuzzy sets is that polar fuzzy sets are described in polar quardinate.

These sets are defined on the universe of angles, hence are repeating shapes every 2π cycles. There are two states for controlling the LFC problem

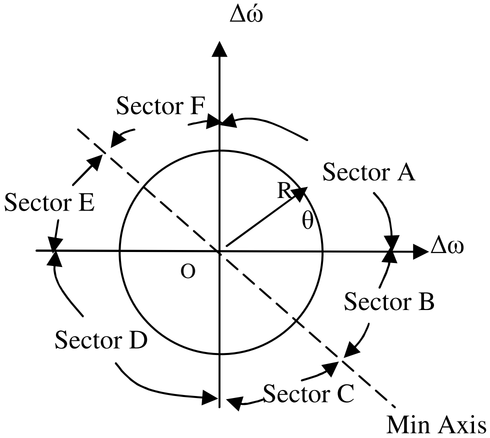


Fig. 2. Six Sector Phase Plane

1. Deviation in system frequency (Δ*f*)
2. Rate of change of system frequency (Δ*ḟ*)

and the calculation of output of PFC is based on angle [14].

In six sector phase plane diagram origin ‘O’ is the desired equilibrium point as shown in Fig. 2. The control action is such that the direction of generator state R always towards origin ‘O’. LN and LP are two fuzzy input membership functions which are define in degree and these two are complimentary to each other where LN stands for large negative and LP for large positive. These two are defined in the range of 45^o to 405^o as shown in Fig. 3.

As described earlier, generation of output of PFC (U_{FLC}) is depending upon the utilization of fuzzy input (angle). P and N are the two triangular membership functions, which are defined in range of -1 to +1 as shown in Fig. 4 where P stands for positive and N for negative.

4.1 Working of PFC

The working of PFC is described in appendix.

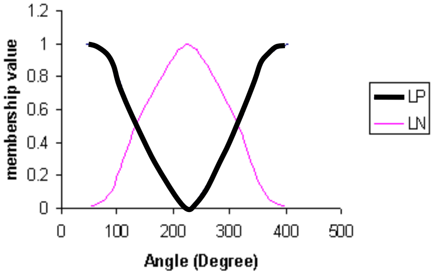


Fig. 3. Fuzzy sets for output variable

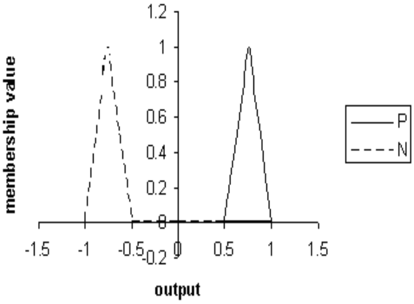


Fig. 4. Fuzzy sets for output variable

5 Application of Polar Fuzzy Logic Controller for LFC

The above developed PFC has been used for load frequency control of single area and two area thermal systems. All studies are for a disturbance of 1% step change in load in one or both areas.

5.1 Load Frequency Control of Single Area System

The quality of supply depends on the voltage and frequency variations. The frequency and the voltage at the generator terminals are determined by the mechanical input and are also affected by the change in load/demand [15]. To control the variations in voltage and frequency, sophisticated controllers detect the load variation and control the input to counter balance the load variation. In conventional control

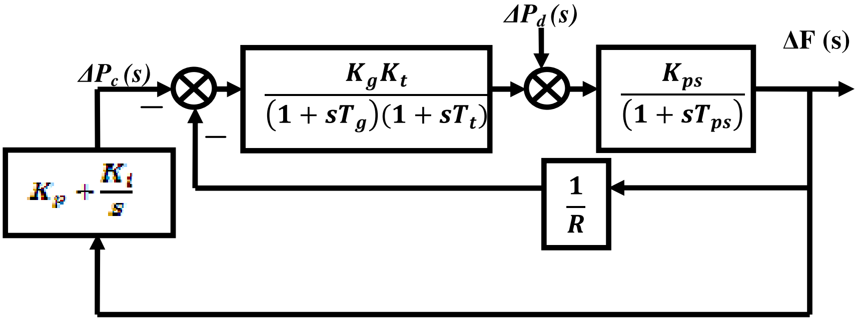


Fig. 5. Block diagram of single area thermal power plant with PI controller

systems, the integral part of the controller controls the frequency variation to zero as early as possible by getting the optimum value of the integral gain Ki.

A small signal model of the single area system is shown in Fig. 5. The turbine output and the electrical load perturbation given in the generator block, gives '$\Delta F(s)$' or 'df' as the output. The single area thermal system described above has been simulated in Matlab 7.0/ Simulink environment.

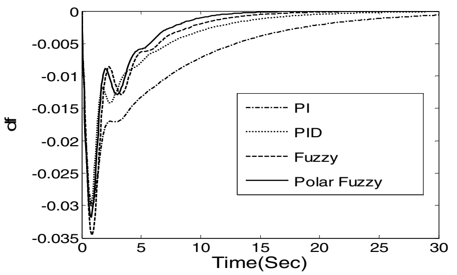


Fig. 6. Frequency variations of single area without Reheat Thermal system

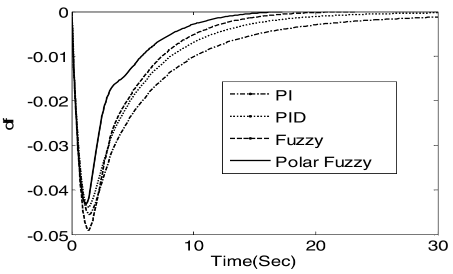


Fig. 7. Frequency variation of single area with Reheat Thermal system

Table 1. Performance comparison of different controllers without and with reheat unit when 1% disturbance in single area thermal system

Controller	Settling Time(sec)		Undershoot(Hz)	
	Without Reheat Unit	With Reheat Unit	Without Reheat Unit	With Reheat Unit
PI	28	30	-0.031	-0.046
PID	18	20	-0.030	-0.044
Fuzzy	13	15	-0.034	-0.049
Polar Fuzzy	10	12	-0.032	-0.043

5.2 Load Frequency Control of Two Area System

When two single areas are tied together through a tie line to maintain their frequency and power exchange as per area contract is called two area system and electric power is transferred through the tie line between the areas [13]. Frequency will deviates from its steady state value if demand in any area is changed. The Simulink model of the two area system has been developed to study the system under different perturbations. With the help of this model system performance can be checked for different controllers for different perturbations. Performance of the PFC has been compared with standard FLC and conventional controllers for with and without reheat unit in the two area thermal system. The results are shown in Figs. 8-10 and Figs. 11-13, respectively, and in Tables 2 and 3.

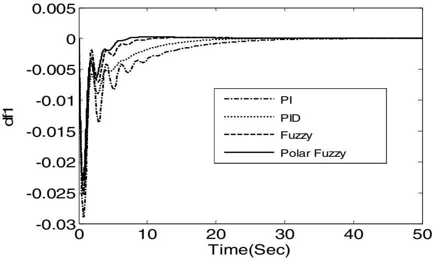


Fig. 8. Frequency variation of area- 1in Two Area Thermal System without Reheat when disturbance in area – 1

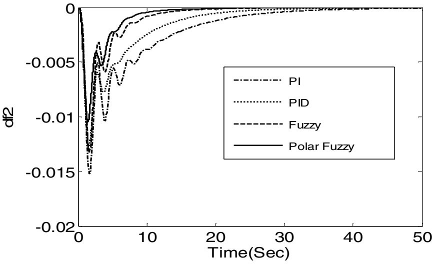


Fig. 9. Frequency variation of area- 2 in a Two Area Thermal System without Reheat when the disturbance in area – 1

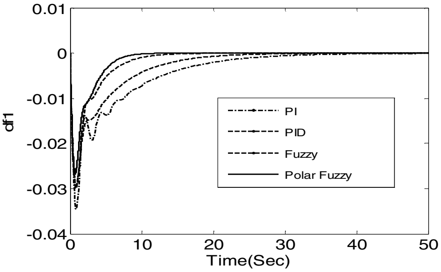


Fig. 10. Frequency variation of area- 1 in a Two Area Thermal System without Reheat when disturbance in both areas

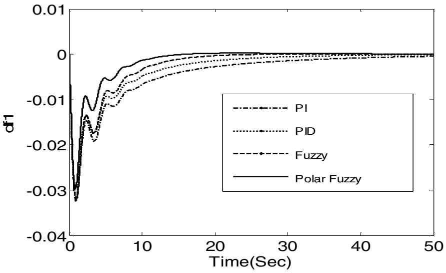


Fig. 11. Frequency variation of area- 1 in a Two Area Thermal System without Reheat when disturbance in area - 1

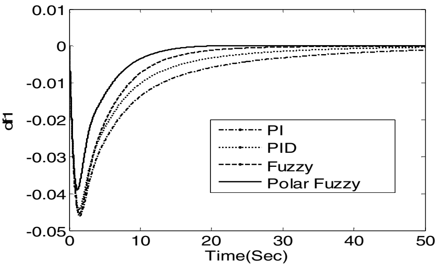


Fig. 12. Frequency variation of area- 2 in a Two Area Thermal System with Reheat unit when disturbance in area - 1

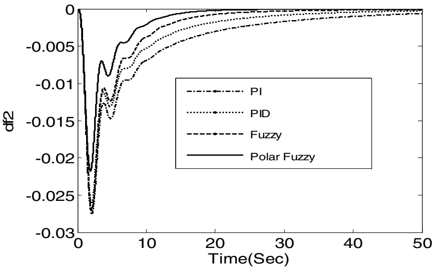


Fig. 13. Frequency variation of area- 1 in a Two Area Thermal System with Reheat unit when disturbance in both areas

Table 2. Performance comparison of different controllers in two area thermal system without reheat unit

1% Disturbance area	Resp. of Area	PI		PID		Fuzzy		Polar Fuzzy	
		Settling Time (Sec)	Under shoot (Hz)	Settling Time (Sec)	Under shoot (Hz)	Settling Time (Sec)	Under shoot (Hz)	Settling Time (Sec)	Under shoot (Hz)
Area 1	Area 1	26	-0.029	20	-0.025	12	-0.025	7	-0.023
Area 1	Area 2	32	-0.015	25	-0.013	15	-0.013	10	-0.010
Both Areas	Area 1	28	-0.035	22	-0.030	13	-0.030	8	-0.027
Both Areas	Area2	28	-0.031	22	-0.030	13	-0.030	8	-0.027

Table 3. Performance comparison of different controllers in two area thermal system with reheat unit

1% Disturbance area	Resp. of Area	PI		PID		Fuzzy		Polar Fuzzy	
		Settling Time (Sec)	Under shoot (Hz)	Settling Time (Sec)	Under shoot (Hz)	Settling Time (Sec)	Under shoot (Hz)	Settling Time (Sec)	Under shoot (Hz)
Area 1	Area 1	43	-0.032	35	-0.032	20	-0.032	13	-0.030
Area 2	Area 2	46	-0.027	35	-0.026	21	-0.027	14	-0.022
Both Areas	Area 1	43	-0.046	35	-0.044	20	-0.045	13	-0.040
Both Areas	Area 2	45	-0.046	35	-0.044	20	-0.045	13	-0.040

6 Co-ordination of PFC and Conventional Controllers of Multi-Area System

The PFC can’t replace all other conventional controllers in a time. This replacement is time taken process over all plants. So it is necessary to study the effect of PFC co-ordination with other conventional controllers. Most of the controllers used in the electrical power system are PI type. Therefore, the effect of the conventional PI and Polar Fuzzy controllers working together needs to be investigated. The comparative performance is studied when the conventional PI is installed in area #1 and the proposed PFC in area #2. Result is compared in Fig. 14 and Tables 4. It can be seen that the performance in terms of settling time and undershoots is much better when PFC is present than with the PI trollers only. The above studies show that the system response, when PFC is present, settles to zero steady state error in less time, with least oscillation and with minimum undershoot compared to the other conventional and Fuzzy logic controllers.

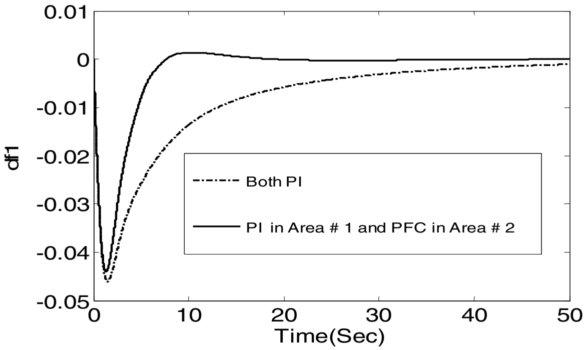


Fig. 14. Frequency variation of area- 1 in Two Area Thermal System with Reheat unit when disturbance in both areas

Table 4. Performance comparison of "Both PI" and co-ordination of "PI with Polar Fuzzy controller" in two area thermal system with reheat

Disturbance	Response of Area	Both PI		Co-ordination of PI with Polar fuzzy	
		Settling Time(Sec)	Undershoot (Hz)	Settling Time(Sec)	Undershoot (Hz)
1% disturbance in Area 1	Area 1	43	-0.032	15	-0.032
	Area 2	46	-0.027	30	-0.025
1% disturbance in Area 2	Area 1	45	-0.028	25	-0.024
	Area 2	45	-0.032	25	-0.030
1% disturbance in both Areas	Area 1	43	-0.046	20	-0.042
	Area 2	45	-0.046	20	-0.040

7 Conclusions

A Polar Fuzzy controller is designed for automatic load frequency control of single area and two area interconnected thermal power systems with and without reheat unit. The controllers performance is compared based on the system settling time and undershoot. A comparison of the system dynamic response with the various controllers shows that the Polar Fuzzy controller yields improved control performance when compared to the PI and PID conventional and standard fuzzy logic controllers.

It is also seen that the PFC performed better than the other conventional PI controller in the comparison study of coordination of PFC and conventional PI controller.

References

[1] Chaturvedi, D.K., Satsangi, P.S., Kalra, P.K.: Load frequency control: a generalized neural network approach. Int. J. on Electric Power and Energy Systems 21(6), 405–415 (1999)

[2] Chaturvedi, D.K.: Electrical Machines Lab Manual with Matlab Programs. University Science Press, New Delhi (2010)

[3] Moon, Y.H., Ryu, H.S., Kim, B., Kim, S., Park, S.C.: Fuzzy logic based extended integral control for load frequency con. IEEE Power Engineering Society Winter Meeting 3, 1289–1293 (2001)

[4] Anand, B., Jeyakumar, A.E.: Load frequency control with fuzzy logic controller considering non-linearities and boiler dyna. ICGST-ACSE Journal 8, 15–20 (2009)

[5] Mathur, H.D., Manjunath, H.V.: Study of dynamic performance of thermal units with asynchronous tie-lines using fuzzy based cont. Journal of Electrical System 3, 124–130 (2007)

[6] Chaturvedi, D.K.: Modeling and Simulation of Systems Using Matlab@/Simulink@. CRC Press, New York (2009)

[7] Das, D., Kothari, M.L., Kothari, D.P., Nanda, J.: Variable structure control strategy to automatic generation control of interconnected reheat thermal systems. Proc. Inst. Elect. Eng. Contr. Theory App. 138(6), 579–585 (1991)

[8] Ogata, K.: Modern control engineering. PHI Publication (2002)

[9] Moon, Y.H., Ryu, H.S., Choi, B.K., Cho, B.H.: Modified PID load frequency control with the consideration of valve position limits. IEEE Power Engineering Society Winter Meeting 1, 701–706 (1999)

[10] Nanda, J., Mangla, A.: Automatic generation control of an interconnected hydro-thermal system using conventional integral and fuzzy logic controller. In: IEEE International Conference on Electric Utility Deregulation, Restructuring and Power Technologies, vol. 1, pp. 372–377 (2004)

[11] Ibraheem, K.P., Kothari, D.P.: Recent philosophies of automatic generation control strategies in power systems. IEEE Transactions on Power Systems 20(1), 346–357 (2005)

[12] Chaturvedi, D.K.: Soft Computing: Applications to electrical engineering problem. Springer, Berlin (2008)

[13] Kundur, P.: Power system stability and control. TMH Publication (2007)

[14] Chaturvedi, D.K., Malik, O.P., Choudhury, U.K.: Polar fuzzy adaptive power system stabilizer. J. of The Institution of Engineers 90, 35–45 (2009)

[15] Nagrath, I.J., Kothari, D.P.: Power System Engineering. TMH Publication (2005)

Appendix: Working of PFC

It's working is described in Chaturvedi (2009) [14]. In Polar Fuzzy logic based controller, there is no need to use two separate input gains for Δf and $\Delta \dot{f}$, because only one input, the polar angle, that depends on the ratio of the properly scaled inputs is used. Thus, only one gain, K_{acc}, is considered. The scaling factor K_{acc} decides as to which variable, frequency deviation or rate of change of frequency has more weight in the magnitude R. In the proposed controller the magnitude of the FLC output is set to be maximum at 45^0, because at this instant the directions of Δf and $\Delta \dot{f}$ are the same and magnitude is also high. On the other hand at 135^0 axis the directions of Δf and $\Delta \dot{f}$ are opposite and, therefore, minimum control action is required. The maximum and minimum is fixed at these angles. But due to the scaling of $\Delta \dot{f}$ with the gain

K_{acc}, all the points in the phase plane are relocated and sometimes system conditions may also require these points to be relocated. Hence, for better tuning of the controller, there is a need for clockwise or anticlockwise rotation. This can be done by adding or subtracting an angle 'β' from phase plane angle 'θ' of the polar form. The polar fuzzy logic controller, shown in Fig. 5, uses angle θ' as input which is defined as:

$$\theta' = (\theta - \beta) + 360^{\circ} \text{ for } \theta - \beta < 45^{\circ}$$

$$\theta' = \theta - \beta \qquad \text{for } \theta - \beta \geq 45^{\circ}.$$

The required control strategy is [refer Fig 2]

a) In sector A ($0^{\circ} - 90^{\circ}$) control from FLC should be large positive as both scaled Δf and $\Delta^{\cdot} f$ are positive.
b) In Sector B (315° - 360°) control signal from FLC should be low positive as scaled Δf is large positive and scaled $\Delta^{\cdot} f$ is small negative.
c) In sector C (270° - 315°) control signal from FLC should be low negative as scaled Δf is small positive and scaled $\Delta^{\cdot} f$ is large negative. In sectors D, E, F all the situations are completely opposite to those in sector A, B and C, respectively. At an angle (θ') of 45° or 405°, the value of membership function of LP is maximum and that for LN is minimum so that 'U_{FLC}' is positive maximum. At angles of 135° and 315° the value of membership function for both LP and LN is the same, so that U_{FLC} is minimum (zero). At an angle of 225^{0} the value of membership function LP is minimum and that for LN is maximum so that U_{FLC} is negative maximum. The working of PFC controller is shown in Fig. 15.

  Output of the fuzzy logic controller is divided into two linguistic variables 'P' and 'N', which are triangular membership functions. So here only two simple rules are considered.
d) Rule 1 - If θ' is LP then U_{FLC} is P.
   Rule 2 - If θ' is LN then U_{FLC} is N.

Hence, the output of FLC unit is $U_{FLC}=f1(\theta, \beta)$,
and final output $u= U_{FLC}* R+ Ko*\Delta f$

Where,

f1 – is a non-linear fuzzy functional mapping;
θ – angle in degree;
R – Magnitude;
β – modifier (tuning parameter) in degree;
Ko – multiplier (tuning parameter);
Δf and $\Delta^{\cdot} f$ – system frequency and rate of change of system frequency

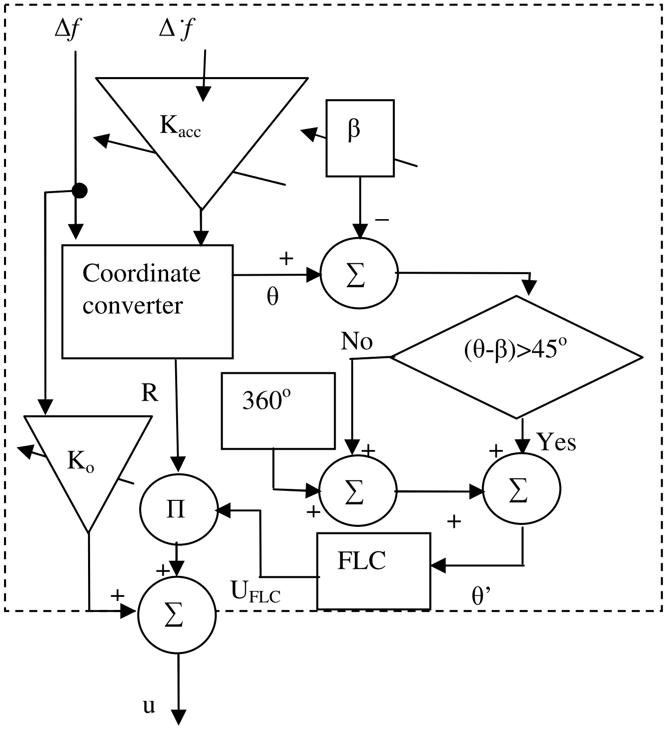


Fig. 15. Block Diagram of Proposed Polar Fuzzy Logic Controller

An Efficient Algorithm to Computing Max-Min Post-inverse Fuzzy Relation for Abductive Reasoning

Sumantra Chakraborty[1], Amit Konar[1], and Ramadoss Janarthanan[2]

[1] Artificial Intelligence Laboratory,
Dept. of Electronics and Tele-Communication Engineering,
Jadavpur University, Calcutta- 32, India
[2] Department of IT, Jaya Engg. College, Chennai, India
`suman_ch54@rediffmail.com`, `konaramit@yahoo.co.in`,
`srmjana_73@yahoo.com`

Abstract. This paper provides an alternative formulation to computing the max-min post-inverse fuzzy relation by minimizing a heuristic (objective) function to satisfy the inherent constraints of the problem. An algorithm for computing the max-min post-inverse fuzzy relation as well as the trace of the algorithm is proposed here. The algorithm exposes its relatively better computational accuracy and higher speed in comparison to the existing technique for post-inverse computation. The betterment of computational accuracy of the max-min post-inverse fuzzy relation leads more accurate result of fuzzy abductive reasoning, because, max-min post-inverse fuzzy relation is required for abductive reasoning.

Keywords: Max–min inverse fuzzy relation, Heuristic Function, Abductive reasoning.

1 Introduction

A fuzzy relation R(x, y) usually describes a mapping from universe X to universe Y (i.e. X →Y), and is formally represented by

$$R(x, y) = \{((x, y), \mu_R(x, y)) \mid (x, y) \in X \times Y\}, \tag{1}$$

where, $\mu_R(x, y)$ denotes the membership of (x, y) to belong to the fuzzy relation R(x, y).

1.1 Fuzzy Max-Min Post-inverse Relation

Let X, Y and Z be three universes and R_1(x, y), for (x, y) ∈ X × Y and R_2(y, z), for (y, z) ∈ Y × Z be two fuzzy relations. Then max-min composition operation of R_1 and R_2, denoted by $R_1 \mathrm{o} R_2$, is a fuzzy relation defined by

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 505–519, 2011.
© Springer-Verlag Berlin Heidelberg 2011

$$R_1 \circ R_2 = \{(x, z), \max_y \{\min\{\mu_{R_1}(x, y), \mu_{R_2}(y, z)\}\}\} \tag{2}$$

where $x \in X$, $y \in Y$ and $z \in Z$.

For brevity we would use '$\wedge$' and '$\vee$' to denote min and max respectively. Thus,

$$R_1 \circ R_2 = \{(x, z), \bigvee_y \{\mu_{R_1}(x, y) \wedge \mu_{R_2}(y, z)\}\}. \tag{3}$$

Let R_1 and R_2 be two fuzzy relational matrices of dimension $(n \times m)$ and $(m \times n)$ respectively. When $R_1 \circ R_2$ = I, the identity relation, we define R_2 as the post-inverse to R_1. It is easy to note that when R_1 = R_2 = I, $R_1 \circ R_2$ = I follows. However, when $R_1 \neq$ I, we cannot find any R_2 that satisfies $R_1 \circ R_2$ = I. It is apparent from the last statements that we can only evaluate approximate max-min post-inverse to R_1, when we know R_1.

1.2 Review

The origin of the proposed max-min fuzzy inverse computation problem dates back to the middle of 1970's, when the researchers took active interest to find a general solution to fuzzy relational equations involving max-min composition operation. The pioneering contribution of solving max-min composition-based relational equation goes to Sanchez [36]. The work was later studied and extended by Prevot [31], Czogala, Drewniak and Pedrycz [8], Lettieri and Liguori [20], Luoh et al. [22], and Yeh [41] for finite fuzzy sets [16]. Cheng-Zhong [7] and Wang et al. [40] proposed two distinct approaches to computing intervals of solutions for each element of an inverse fuzzy relation. Higashi and Klir in [14] introduced a new approach to computing maximal and minimal solutions to a fuzzy relational equation. Among the other well-known approaches to solve fuzzy relational equations, the works presented in [10], [12], [13], [15], [29], [43], [44], [30], [21] need special mention. The early approaches mentioned above unfortunately is not directly applicable to find general solutions R_2, satisfying the relational equation: $R_1 \circ R_2$ = I. Interestingly, there are problems like fuzzy backward/abductive [17] reasoning, where R_1 $\neq$ I, but R_2 needs to be evaluated. This demands a formulation to determine a suitable R_2, such that the given relational equation is *best satisfied*.

Several direct (or indirect) formulations of the *max-min* pre-inverse computing problem have been addressed in the literature [2], [6], [7], [24], [25], [26], [27], [34], [35], [37], [45]. A first attempt to compute fuzzy pre-inverse with an aim to satisfy all the underlying constraints in the relational equation using a heuristic objective function is addressed in [35]. The work, however, is not free from limitations as the motivations to optimize the heuristic (objective) function to optimally satisfy all the constraints are not fully realized. All the limitations in [35] are fully resolved by [45].

This paper is an extended part of [45], where the proposed work illustrates an alternative formulation to computing the max-min post-inverse fuzzy relation.

The rest of the paper is organized as follows. In section 2, we provide *Strategies* used to solve the post-inverse computational problem. The algorithm is presented in section 3. The analysis of algorithm and example is given section 3.1 and 3.2 respectively.

2 Proposed Computational and Approach to Fuzzy Max-Min Post-inverse Relation

Given a fuzzy relational matrix R of dimension $(m \times n)$, we need to evaluate a Q matrix of dimension $(n \times m)$ such that R o Q = I′ ≈ I, where I denotes identity matrix of dimension $(m \times m)$. Let q_j be the j^{th} column of Q matrix. The following strategies have been adopted to solve the equation R o Q = I′ ≈ I for known R.

Strategy 1: Decomposition of R o Q ≈ I into [R o q_j $]_j \approx 1$ and [R o q_j $]_{l,l\neq j} \approx 0$.

Since, R o Q ≈ I, R o $q_j \approx j^{\text{th}}$ column of I matrix, therefore, the j^{th} element of R o q_j, denoted by [R o q_j $]_j \approx 1$ and the l^{th} element (where $l \neq j$) of R o q_j, denoted by [R o q_j $]_{l,l\neq j} \approx 0$.

Strategy 2: Determination of the effective range of q_{ij}, $\forall i$ in $[0, r_{ji}]$.

Since Q is a fuzzy relational matrix, its elements $q_{ij} \in [0, 1]$ for $\forall i, j$. However, to satisfy the constraint [R o q_j $]_j \approx 1$, the range of q_{ij}, $\forall i$ virtually becomes $[0, r_{ji}]$ by Lemma 1.

This range is hereafter referred to as *effective range* of q_{ij}.

Lemma 1: The constraint [R o q_j $]_j \approx 1$, sets the effective range of q_{ij} in $[0, r_{ji}]$.

Proof: [R o q_j $]_j$

$$= \bigvee_{i=1}^{n} (r_{ji} \wedge q_{ij}) \tag{4}$$

Since $[\bigvee_{i=1}^{n} (r_{ji} \wedge q_{ij})]_{q_{ij} > r_{ji}} = [\bigvee_{i=1}^{n} (r_{ji} \wedge q_{ij})]_{q_{ij} = r_{ji}}$, The minimum value of q_{ij} that maximizes [R o q_j $]_j$ towards 1 is r_{ji}. Setting q_{ij} beyond r_{ji} is of no use in connection with maximization of [R o q_j $]_j$ towards 1. Therefore, the effective range of q_{ij} reduces from [0, 1] to [0, r_{ji}]. □

Strategy 3: Replacement of the constraint [R o q_j $]_j \approx 1$, by $q_{kj} \approx 1$, where $q_{kj} \geq q_{ij}$, $\forall i$

We first prove [R o q_j $]_j = q_{kj}$ for $q_{kj} \geq q_{ij}$, $\forall i$ by Lemma 2, and then argue that [R o q_j $]_j \approx 1$ can be replaced by $q_{kj} \approx 1$.

Lemma 2: If $q_{kj} \geq q_{ij}$, $\forall i$, then [R o q_j $]_j = q_{kj}$.

Proof: $$[R \circ q_j]_j = \bigvee_{i=1}^{n} (r_{ji} \wedge q_{ij}) \tag{5}$$

By Lemma 1, we can write $0 \leq q_{ij} \leq r_{ji}$, $\forall i$.

Therefore, $$(r_{ji} \wedge q_{ij}) = q_{ij} \tag{6}$$

Substituting expression (6) in expression (5), yields the resulting expression as

$$[R \circ q_j]_j = \bigvee_{i=1}^{n} (q_{ij}) \tag{7}$$

$$= q_{kj} \quad \text{as } q_{kj} \geq q_{ij}, \forall i. \tag{8}$$

□

The maximization of [R o q_j $]_j$, therefore, depends only on q_{kj}, and the maximum value of [R o q_j $]_j = q_{kj}$. Consequently, the constraint [R o q_j $]_j \approx 1$ is replaced by $q_{kj} \approx 1$. Discussion on Strategy 3 ends here. A brief justification to strategy 4-6 is outlined next.

Justification of Strategies 4 to 6: In this paper, we evaluate the largest element q_{kj} and other element q_{ij} (for $i \neq k$) in q_j, the j^{th} column of Q-matrix, by separate procedures. For evaluation of q_{kj}, we first need to identify the positional index k of q_{kj} so that maximization of [R o q_j $]_j$ and minimization of [R o q_j $]_{l, l \neq j}$ occur jointly for a suitable selection of q_{kj}. This is taken care of in Strategy 5 and 6. In Strategy 5, we determine k for the possible largest element q_{kj}, whereas in Strategy 6 we evaluate q_{kj}. To determine q_{ij} (for $i \neq k$), we only need to minimize $[R \circ q_j]_{l, l \neq j}$. This is considered in Strategy 4. It is indeed important to note that selection of q_{ij} (for $i \neq k$) to minimize $[R \circ q_j]_{l, l \neq j}$ does not hamper maximization of $[R \circ q_j]_j$ as $[R \circ q_j]_j = q_{ik}$, vide Lemma 2.

Strategy 4: Evaluation of q_{ij}, $i \neq k$, *where* $q_{kj} \geq q_{ij}$, $\forall i$.

The details of the above strategy are taken up in Theorem 1.

Theorem 1: If $q_{kj} \geq q_{ij}$, $\forall i$, *then the largest value of* $q_{ij}\Big|_{i \neq k}$ *that minimizes* $[R \circ q_j]_{l, l \neq j}$ *towards 0 is given by* $(\bigwedge_{\substack{l=1 \\ l \neq j}}^{m} r_{lk}) \wedge q_{kj}$.

Proof: $[R \circ q_j]_{l, l \neq j}$

$$= \bigvee_{i=1}^{n} (r_{li} \wedge q_{ij}), \quad \forall\, l, l \neq j \tag{9}$$

$$= \bigvee_{\substack{i=1 \\ i \neq k}}^{n} (r_{li} \wedge q_{ij}) \vee (r_{lk} \wedge q_{kj}), \quad \forall\, l, l \neq j \tag{10}$$

$$= (r_{lk} \wedge q_{kj}) \quad \forall\, l, l \neq j,$$

$$\text{if } (r_{lk} \wedge q_{kj}) \geq \bigvee_{\substack{i=1 \\ i \neq k}}^{n} (r_{li} \wedge q_{ij}). \tag{11}$$

Therefore,

$$Min\, [R \circ q_j]_{l, l \neq j} = \underset{\forall l, l \neq j}{Min} (r_{lk} \wedge q_{kj})$$

$$= \underset{\forall l, l \neq j}{Min} \{r_{lk}\} \wedge q_{kj}$$

$$= (\bigwedge_{\substack{l=1 \\ l \neq j}}^{m} r_{lk}) \wedge q_{kj} \tag{12}$$

Since $Min[R \circ q_j]_{l,l\neq j} = (\bigwedge_{\substack{l=1\\ l\neq j}}^{m} r_{lk}) \wedge q_{kj}$ and the largest value in $[R \circ q_j]_{l,l\neq j} = (r_{lk} \wedge q_{kj})$, therefore, $Min[R \circ q_j]_{l,l\neq j}$, will be the largest among $(r_{li} \wedge q_{ij})\, \forall\, i,\, i \neq k$, if

$$Min[R \circ q_j]_{l,l\neq j} = (\bigwedge_{\substack{l=1\\ l\neq j}}^{m} r_{lk}) \wedge q_{kj}$$

$$\geq \bigvee_{\substack{i=1\\ i\neq k}}^{n} (r_{li} \wedge q_{ij}), \tag{13}$$

which is same as, $(\bigwedge_{\substack{l=1\\ l\neq j}}^{m} r_{lk}) \wedge q_{kj} \geq (r_{li} \wedge q_{ij}) \quad \forall\, i,\, i \neq k.$ (14)

The *largest value of* q_{ij} for $i \neq k$ can be obtained by setting equality in equation (14), and the resulting equality condition is satisfied when

$$q_{ij}\Big|_{i\neq k} = (\bigwedge_{\substack{l=1\\ l\neq j}}^{m} r_{lk}) \wedge q_{kj}\,. \tag{15}$$

□

Strategy 5: Determining the positional index k for the element q_{kj} *(* $\geq q_{ij}$*,* $\forall j$ *) in* q_j.

To determine the position k of q_{kj} in q_j, we first need to construct a heuristic function $h(q_{kj})$ that satisfies two constraints:

i) Maximize $[R \circ q_j]_j$ (16)

ii) Minimize $[R \circ q_j]_{l,l\neq j}$, (17)

and then determine the index k, such that $h(q_{kj}) \geq h(q_{ij})$, $\forall i$. In other words, we need to determine the positional index k for the possible largest element q_{kj} in the j^{th}

column of Q-matrix, as the largest value of $h(q_{kj})$ ensures maximization of the heuristic function $h(q_{kj})$, and thus best satisfies the constraints (16) and (17).

Formulation of the heuristic function is considered first, and the determination of k satisfying $h(q_{kj}) \geq h(q_{ij})$, $\forall i$ is undertaken next.

One simple heuristic cost function that satisfies (16) and (17) is

$$h_1(q_{kj}) = q_{kj} - \frac{1}{(m-1)} \sum_{\substack{l=1 \\ l \neq j}}^{m} (r_{lk} \wedge q_{kj}),$$

where $q_{kj} \geq q_{ij}$, $\forall i$, by Theorem 2. Next we find k such that

$$\underset{q_{kj} \in [0, r_{jk}]}{Max} h_1(q_{kj}) \geq \underset{q_{ij} \in [0, r_{ji}]}{Max} h_1(q_{ij}), \text{ for all } i.$$

Theorem 2: If $q_{kj} \geq q_{ij}$, $\forall i$, *then maximization of* $[R \text{ o } q_j]_j$ *and minimization of* $[R \text{ o } q_j]_{l, l \neq j}$ *can be represented by a heuristic function,*

$$h_1(q_{kj}) = q_{kj} - \frac{1}{(m-1)} \sum_{\substack{l=1 \\ l \neq j}}^{m} (r_{lk} \wedge q_{kj}).$$

Proof: Given $q_{kj} \geq q_{ij}$, for all i, Thus from Lemma 2, we have

$$[R \text{ o } q_j]_j = q_{kj} \tag{18}$$

Further, $[R \text{ o } q_j]_{l, l \neq j}$

$$= (r_{l1} \wedge q_{1j}) \vee (r_{l2} \wedge q_{2j}) \vee \ldots \vee (r_{lk} \wedge q_{kj}) \vee \ldots \vee (r_{ln} \wedge q_{nj}), \text{ for } \forall l,\ l \neq j. \tag{19}$$

Now by Theorem 1 we have $q_{ij} \Big|_{i \neq k} = (\bigwedge_{\substack{l=1 \\ l \neq j}}^{m} r_{lk}) \wedge q_{kj}$ and substituting this value in equation (19) we have

$$[R \text{ o } q_j]_{l, l \neq j} = (r_{lk} \wedge q_{kj}), \text{ for } \forall l,\ l \neq j \tag{20}$$

Now to jointly satisfy maximization of $[R \text{ o } q_j]_j$ and minimization of $[R \text{ o } q_j]_{l, l \neq j}$ we design a heuristic function,

$h_1(q_{kj}) = [\mathrm{R} \circ q_j\,]_j - \mathrm{f}((r_{1k} \wedge q_{kj}),\ldots,(r_{j-1,k} \wedge q_{kj}), (r_{j+1,k} \wedge q_{kj}), \ldots\ldots,$ $(r_{mk} \wedge q_{kj}))$.

Now, for any monotonically non-decreasing function f(·), maximization of $[\mathrm{R} \circ q_j\,]_j$ and minimization of the *min* terms $(r_{1k} \wedge q_{kj})$, ……., $(r_{j-1,k} \wedge q_{kj})$, $(r_{j+1,k} \wedge q_{kj})$, ……., $(r_{mk} \wedge q_{kj})$ calls for maximization of $h_1(q_{kj})$. Since averaging (Avg) is a monotonically increasing function, we replace f(·) by Avg(·). Thus,

$$h_1(q_{kj}) = [\mathrm{R} \circ q_j\,]_j - \mathrm{Avg}((r_{1k} \wedge q_{kj}),\ldots,(r_{j-1,k} \wedge q_{kj}), (r_{j+1,k} \wedge q_{kj}), \ldots\ldots,(r_{mk} \wedge q_{kj}))$$

$$= q_{kj} - \frac{1}{(m-1)} \sum_{\substack{l=1 \\ l \neq j}}^{m} (r_{lk} \wedge q_{kj}). \tag{21}$$

Although apparent, it may be added for the sake of completeness that $n>1$ in (21). □

The determination of index *k*, such that $h_1(q_{kj}) \geq h_1(q_{ij})$, $\forall i$ can be explored now. Since $q_{ij} \in [0,\ r_{ji}\,]$ and $q_{kj} \in [0,\ r_{jk}\,]$, therefore $h_1(q_{kj}) \geq h_1(q_{ij})$, $\forall i$ can be transformed to

$$\underset{q_{kj} \in [0, r_{jk}]}{Max}\ h_1(q_{kj}) \geq \underset{q_{ij} \in [0, r_{ji}]}{Max}\ h_1(q_{ij}), \text{ for all } i \tag{22}$$

Consequently, determination of *k* satisfying the inequality (22) yields the largest element q_{kj} in the q_j, that maximizes the heuristic function $h_1(.)$.

Corollary 1: $h_2(q_{kj}) = q_{kj} - \sum_{\substack{l=1 \\ l \neq j}}^{m} (r_{lk} \wedge q_{kj})$, *too is a heuristic function that maximizes* $[R \circ q_j\,]_j$ *and minimizes* $[R \circ q_j\,]_{l, l \neq j}$.

Proof: Since $\sum_{\substack{l=1 \\ l \neq j}}^{m} (r_{lk} \wedge q_{kj})$ is a monotonically non-decreasing function, thus the Corollary can be proved similar to Theorem 2. □

Strategy 6: Finding the maximum value of $h_1(q_{ij})$ *for* q_{ij} *in [0,* r_{ji} *].*

We first of all prove that $h_1(q_{ij})$ is a monotonically non-decreasing function of q_{ij} by Theorem 3. Then we can easily verify that for q_{ij} in [0, r_{ji}], $h_1(q_{ij})\Big|_{q_{ij}=r_{ji}}$ is the largest, i.e., $h_1(q_{ij})\Big|_{q_{ij}=r_{ji}} \geq h_1(q_{ij})\Big|_{q_{ij}\in[0,r_{ji}]}$.

Theorem 3: $h_1(q_{ij}) = q_{ij} - \dfrac{1}{(m-1)}\sum_{\substack{l=1\\l\neq j}}^{m}(r_{li}\wedge q_{ij})$*, is a monotonically non-decreasing function of* q_{ij} *in [0,* r_{ji} *].*

Proof: We consider two possible cases:

Case 1: If $q_{ij} > r_{li}$ $\forall$ *l, l* ≠ *j*, then,

$$h_1'(q_{ij}) = \frac{dh_1(q_{ij})}{dq_{ij}} = 1 - \frac{1}{(m-1)}\sum_{\substack{l=1\\l\neq i}}^{m}\frac{d(r_{li})}{dq_{ij}} = 1\ (>0). \qquad \text{(since, } \frac{d(r_{li})}{dq_{ij}} = 0\text{).}$$

Case 2: Let $q_{ij} \leq r_{li}$ for at least one *l* (say *t* times), then,

$$\begin{aligned} h_1'(q_{ij}) &= \frac{dh_1(q_{ij})}{dq_{ij}} \\ &= 1 - \frac{1}{(m-1)}\frac{d}{dq_{ij}}\sum_{t-times}(q_{ij}) \\ &= 1 - \frac{t}{(m-1)} \geq 0 \quad \text{as } t \leq (m-1). \end{aligned}$$

$\therefore$ $h_1'(q_{ij}) \geq 0$, and therefore, $h_1(\cdot)$ is a monotonically non-decreasing function of q_{ij} in [0, r_{ji}]. □

Strategy 7: Finding Q-matrix

Evaluation of each row q_j of Q-matrix is considered independently. For given row q_j, we need to determine the largest element $q_{kj} \geq q_{ij}$, $\forall i$. After q_{kj} is evaluated, we evaluate q_{ij}, $i \neq k$ in the subsequent phase.

Since maximization of $h_1(\cdot)$ ensures satisfaction of the constraints (16) and (17) in strategy 5, to determine q_{kj}, we look for an index *k*, such that

$$\underset{q_{kj}\in[0,r_{jk}]}{Max} h_1(q_{kj}) \geq \underset{q_{ij}\in[0,r_{ji}]}{Max} h_1(q_{ij}), \text{ for all } i.$$

Further, since $h_1(\cdot)$ is a monotonically non-decreasing function, above inequality reduces to,

$$h_1(q_{kj})\Big|_{q_{kj}=r_{jk}} \geq h_1(q_{ij})\Big|_{q_{ij}=r_{ji}} \tag{23}$$

If a suitable value of k is found satisfying (23), we say q_{ik} is the largest element in q_i and the value of $q_{kj} = r_{jk}$.

For other element q_{ij} in q_j, $i \neq k$, we evaluate

$$q_{ij}\Big|_{i\neq k} = \Big(\bigwedge_{\substack{l=1\\ l\neq j}}^{m} r_{lk}\Big) \wedge q_{kj}$$

$$= \Big(\bigwedge_{\substack{l=1\\ l\neq j}}^{m} r_{lk}\Big) \wedge r_{jk} \qquad \text{(Since, } q_{kj} = r_{jk}\text{)}$$

$$= \bigwedge_{l=1}^{m} r_{lk}. \tag{24}$$

The principle for evaluation of q_{ij} for a given j can be repeated for j = 1 to m to determine Q-matrix.

3 Proposed Fuzzy Max-Min Post-inverse Computation Algorithm

The results obtained from the strategies in Section 2 are used here to construct Algorithm for Max-Min Post-inverse computation for a given fuzzy relational matrix.

```
ALGORITHM 1
Input: R = $[r_{ij}]_{m\times n}$ where $0 \leq r_{ij} \leq 1$ $\forall i,j$;
Output: Q = $[q_{ij}]_{n\times m}$ where $0 \leq q_{ij} \leq 1$ $\forall i, j$;
                                        //such that RoQ is close enough to I//
 Begin
     For j = 1 to m
```

```
            Evaluate $q_{kj}$ (.);        //Determine the position k and value of the largest
                                          //element $q_{kj}$ in column j of Q.//
            For $i$ = 1 to n
               If( $i \neq k$ )
                  Then $q_{ij} = \underset{l}{Min}\{ r_{lk} \}$;
                                    //Determine all the elements in the $j^{th}$ column of Q matrix
                                    //except $q_{kj}$ //
               End If;
            End For;
      End For;
   End.

   Evaluate $q_{kj}$ (.)
   Begin
      For $i$ = 1 to n
         $h_1(q_{ij}) = r_{ji} - \frac{1}{(m-1)} \sum_{l=1, l \neq j}^{m} (r_{li} \wedge r_{ji})$;   //Evaluate $h_1(q_{ij})$. //
      End For
      If $h_1(q_{kj}) \geq h_1(q_{ij})$ for all $i$
      Then return k and $q_{kj} = r_{jk}$;   //Return the position k of $q_{kj}$, and its value//
   End.
```

3.1 Explanation of Algorithm I

Algorithm 1 evaluates the elements in q_j, i.e., $q_{1j}, q_{2j}, \ldots\ldots, q_{mj}$ in a single pass by determining the position k of the largest element q_{kj} in the j^{th} column and then its value r_{jk}. Next, we determine the other elements in q_j, which is given by

$$q_{ij}\Big|_{i \neq k} = \bigwedge_{l=1}^{m} r_{lk} \,.$$

The outer **For**-loop in the algorithm sets j = 1 to m with an increment in j by 1, and evaluation of q_j takes place for each setting of j. The most important step inside this outer **For**-loop is determining positional index k of the largest element q_{kj} in q_j and evaluation of its value. This has been taken care of in function Evaluate q_{kj} (.).

Table 1. Trace of the algorithm

$j=1$

Evaluate q_{kj} (j)

When $q_{11}=r_{11}=0.9$; $h(q_{11}) = r_{11}-\{(r_{21}\wedge r_{11})+(r_{31}\wedge r_{11})\}/2 = 0.8$
When $q_{21}=r_{12}=0.4$; $h(q_{21}) = r_{12}-\{(r_{22}\wedge r_{12})+(r_{32}\wedge r_{12})\}/2 = 0.05$
When $q_{31}=r_{13}=0.3$; $h(q_{31}) = r_{13}-\{(r_{23}\wedge r_{13})+(r_{33}\wedge r_{13})\}/2 = 0.05$

Max{ $h(q_{11})$, $h(q_{21})$, $h(q_{31})$ }= $h(q_{11})$; Return $q_{11}=0.9$ and k=1

Evaluate the largest q_{ij} for all i except k =1

$q_{21}=\mathrm{Min}(r_{11}, r_{21}, r_{31}) = 0.1$; $q_{31}=\mathrm{Min}(r_{11}, r_{21}, r_{31}) = 0.1$;

$j=2$

Evaluate_ q_{kj} (j)

When $q_{12}=r_{21}=0.1$; $h(q_{12}) = r_{21}-\{(r_{11}\wedge r_{21})+(r_{31}\wedge r_{21})\}/2 = 0.0$
When $q_{22}=r_{22}=0.3$; $h(q_{22}) = r_{22}-\{(r_{12}\wedge r_{22})+(r_{32}\wedge r_{22})\}/2 = 0.0$
When $q_{32}=r_{23}=0.8$; $h(q_{32}) = r_{23}-\{(r_{13}\wedge r_{23})+(r_{33}\wedge r_{23})\}/2 = 0.55$

Max{ $h(q_{12})$, $h(q_{22})$, $h(q_{32})$ }= $h(q_{32})$; Return $q_{32} = 0.8$ and k=3

Evaluate the largest q_{ij} for all i except k = 3

$q_{12} = \mathrm{Min}(r_{13}, r_{23}, r_{33}) = 0.2$; $q_{22} = \mathrm{Min}(r_{13}, r_{23}, r_{33}) = 0.2$;

$j=3$

Evaluate_ q_{kj} (j)

When $q_{13}=r_{31}=0.1$; $h(q_{13}) = r_{31}-\{(r_{11}\wedge r_{31})+(r_{21}\wedge r_{31})\}/2 = 0.0$
When $q_{23}=r_{32}=1.0$; $h(q_{23}) = r_{32}-\{(r_{12}\wedge r_{32})+(r_{22}\wedge r_{32})\}/2 = 0.65$
When $q_{33}=r_{33}=0.2$; $h(q_{33}) = r_{33}-\{(r_{13}\wedge r_{33})+(r_{23}\wedge r_{33})\}/2 = 0.0$

Max{ $h(q_{13})$, $h(q_{23})$, $h(q_{33})$ }= $h(q_{23})$; Return $q_{23}=1.0$ and k=2

Evaluate the largest q_{ij} for all i except k = 2

$q_{13} = \mathrm{Min}(r_{12}, r_{22}, r_{32}) = 0.3$; $q_{33} = \mathrm{Min}(r_{12}, r_{22}, r_{32}) = 0.3$;

3.2 Example

Given $R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} 0.9 & 0.4 & 0.3 \\ 0.1 & 0.3 & 0.8 \\ 0.1 & 1.0 & 0.2 \end{bmatrix}$, we now provide a trace of the Algorithm in Table-1.

$$\therefore \quad Q = \begin{bmatrix} 0.9 & 0.2 & 0.3 \\ 0.1 & 0.2 & 1.0 \\ 0.1 & 0.8 & 0.3 \end{bmatrix}.$$

References

1. Arnould, T., Tano, S.: Interval-valued fuzzy backward reasoning. IEEE Trans. Fuzzy Systems 3(4), 425–437 (1995)
2. Arnould, T., Tano, S.: The inverse problem of the aggregation of fuzzy sets. Int. J. Uncertainity Fuzzyness Knowledge Based System 2(4), 391–415 (1994)
3. Arnould, T., Tano, S., Kato, Y., Miyoshi, T.: Backward chaining with fuzzy "if..then...rules". In: Proc. of the Second IEEE International Conference on Fuzzy Systems, San Francisco, CA, pp. 548–533 (1993)
4. Bender, E.A.: Mathematical Methods in Artificial Intelligence. IEEE Computer Society Press Silver Spring, MD (1996)
5. Bourke, M.M., Fisher, D.G.: Solution algorithms for fuzzy relational equations with max-product composition. Fuzzy Sets and Systems 94, 61–69 (1998)
6. Cen, J.: Fuzzy matrix partial ordering and generalized inverses. Fuzzy Sets and Systems 105, 453–458 (1999)
7. Cheng-Zhong, L.: Generalized inverses of fuzzy matrix. Approximate Reasoning in Decis. Anal., 57–60 (1982)
8. Czogala, E., Drewniak, J., Pedrycz, W.: Fuzzy relation equations on a finite set. Fuzzy Sets and Systems 12, 89–101 (1982)
9. Di Nola, A., Sessa, S.: On the set of solutions of composite fuzzy relation equations. Fuzzy Sets and Systems 9, 275–285 (1983)
10. Drewniak, J.: Fuzzy relation inequalities. Cybernet. Syst. 88, 677–684 (1988)
11. Drewniak, J.: Fuzzy relation equations and inequalities. Fuzzy Sets and Systems 14, 237–247 (1984)
12. Gottwald, S.: Generalized solvability criteria for fuzzy equations. Fuzzy Sets and Systems 17, 285–296 (1985)
13. Gottwald, S., Pedrycz, W.: Solvability of fuzzy relational equations and manipulation of fuzzy data. Fuzzy Sets and Systems 18, 45–65 (1986)
14. Higashi, M., Klir, G.J.: Resolution of finite fuzzy relation equations. Fuzzy Sets and Systems 13, 65–82 (1984)
15. Kim, K.H., Roush, F.W.: Generalized fuzzy matrices. Fuzzy Sets and Systems 4, 293–315 (1980)
16. Klir, G.J., Folger, T.A.: Fuzzy Sets, Uncertainity, and Information, ch. 1. Prentice-Hall, Englewood Cliffs (1988)

17. Konar, A.: Computational Intelligence: Principles, Techniques and Applications. Springer, Heidelberg (2005)
18. Kosko, B.: Fuzzy Engineering. Prentice-Hall, Englewood Cliffs (1997)
19. Leotamonphong, J., Fang, S.-C.: An efficient solution procedure for fuzzy relational equations with max-product composition. IEEE Trans. Fuzzy Systems 7(4), 441–445 (1999)
20. Lettieri, A., Liguori, F.: Characterization of some fuzzy relation equations provided with one solution in a finite set. Fuzzy Sets and Systems 13, 83–94 (1984)
21. Li, X., Ruan, D.: Novel neural algorithms based on fuzzy δ -rules for solving fuzzy relational equations, part III. Fuzzy Sets and Systems 109, 355–362 (2000)
22. Luoh, L., Wang, W.-J., Liaw, Y.-K.: Matrix-pattern-based computer algorithm for solving fuzzy relational equations. IEEE Trans. on Fuzzy Systems 11(1) (February 2003)
23. Miyakoshi, M., Shimbo, M.: Solutions of composite fuzzy relational equations with triangular norms. Fuzzy Sets and Systems 16, 53–63 (1985)
24. Pappis, C.P.: Multi-input multi-output fuzzy systems and the inverse problem. European J. of Operational Research 28, 228–230 (1987)
25. Pappis, C.P., Adamopoulos, G.I.: A computer algorithm for the solution of the inverse problem of fuzzy systems. Fuzzy Sets and Systems 39, 279–290 (1991)
26. Pappis, C.P., Sugeno, M.: Fuzzy relational equation and the inverse problem. Fuzzy Sets and Systems 15(1), 79–90 (1985)
27. Pedrycz, W.: Inverse problem in fuzzy relational equations. Fuzzy Sests and Systems 36, 277–291 (1990)
28. Pedrycz, W.: Fuzzy Set Calculus. In: Ruspini, E.H., Bonissone, P.P., Pedrycz, W. (eds.) Handbook of Fuzzy Computation. IOP Publishing, Bristol (1998)
29. Pedrycz, W.: Fuzzy relational equations with generalized connectives and their applications. Fuzzy Sets and Systems 10, 185–201 (1983)
30. Perfilieva, I., Tonis, A.: Compatibility of systems of fuzzy relation equations. Int. J. General Systems 29(4), 511–528 (2000)
31. Prevot, M.: Algorithm for the solution of fuzzy relations. Fuzzy Sets and Systems 5, 319–322 (1981)
32. Rich, E., Knight, K.: Artificial Intelligence. McGraw-Hill, New York (1991)
33. Ross, T.J.: Fuzzy Logic with Engg. Applications. McGraw-Hill, New York (1995)
34. Saha, P.: Abductive Reasoning with Inverse fuzzy discrete Relations, Ph. D. Dissertation, Jadavpur University, India (2002)
35. Saha, P., Konar, A.: A heuristic algorithm for computing the max-min inverse fuzzy relation. Int. J. of Approximate Reasoning 30, 131–147 (2002)
36. Sanchez, E.: Resolution of composite fuzzy relation equations. Inf. Control 30, 38–48 (1976)
37. Sanchez, E.: Solution of fuzzy equations with extended operations. Fuzzy Sets and Systems 12, 237–247 (1984)
38. Sessa, S.: Some results in the setting of fuzzy relation equation theory. Fuzzy Sets and Systems 14, 281–297 (1984)
39. Togai, M.: Application of fuzzy inverse relation to synthesis of a fuzzy controller for dynamic systems. In: Proc. 23rd Conf. on Decision and Control, Las Vegs, NV (December 1984)
40. Wang, S., Fang, S.-C., Nuttle, H.L.W.: Solution set of interval valued fuzzy relational equation. Fuzzy Optim. Decision Making 2(1), 41–60 (2003)

41. Wu, Y.-K., Guu, S.-M.: An efficient procedure for solving a fuzzy relational equation with Max-Archimedean t-norm composition. IEEE Trans. on Fuzzy Systems 16(1) (February 2008)
42. Yeh, C.-T.: On the minimal solutions of max-min fuzzy relational equations. Fuzzy Sets and Systems 159(1) (January 2008)
43. Zadeh, L.A.: Knowledge representation in fuzzy logic. IEEE Trans. Knowledge Data Engg. (1988)
44. Zimmerman, H.J.: Fuzzy Set Theory and its Applications. Kluwer Academic Publishers, Dordrecht (1996)
45. Chakraborty, S., Konar, A., Jain, L.C.: An Efficient Algorithm to Computing Max-Min Inverse Fuzzy Relation for Abductive Reasoning. IEEE Trans. Sys. Man Cybernet (A) 40(1) (January 2010)

Fuzzy-Controlled Energy-Efficient Weight-Based Two Hop Clustering for Multicast Communication in Mobile Ad Hoc Networks

Anuradha Banerjee[1], Paramartha Dutta[2], and Subhankar Ghosh[3]

[1] Kalyani Govt. Engg. College, Kalyani, Nadia
West Bengal, India
anuradha79bn@gmail.com
[2] Visva-Bharati University, Santiniketan, West Bengal, India
paramartha.dutta@gmail.com
[3] Regent Education and Research Foundation, West Bengal, India
redhatsubha@gmail.com

Abstract. A mobile ad hoc network is an infrastructure-less network where the nodes are free to move independently in any direction. The technique of clustering is used to divide a large network into several sub-networks. Inherent dynamism of nodes gives rise to unpredictable topological structure in ad hoc networks which complicates the clustering process. Moreover, since mobile nodes are battery powered, energy saving is required to increase node lifetime and maintain network connectivity. In this paper we propose a fuzzy-controlled energy-efficient weight-based clustering algorithm (FEW-CA) for mobile ad hoc networks. A fuzzy controller named Cluster-head Elector (CE) is embedded in each node n_i which computes weight of node n_i. If n_i is not already member of any cluster and its weight is greater than a pre-defined threshold, then it is elected as a cluster-head and a cluster is formed with 1-hop and 2-hop neighbors of n_i who are not already members of any other cluster. FEW-CA does not allow any node to become member of more then one cluster simultaneously. Simulation results firmly establish that our algorithm performs better than existing state-of-the-art broadcast algorithms.

Keywords: Ad hoc network, Cluster, Energy-efficiency, Fuzzy.

1 Introduction

Various clustering algorithms for ad hoc networks appear in the literature. Among them, Highest Degree (HD) [1] algorithm, Lowest ID heuristics (LID) [2], Distributed Clustering Algorithm (DCA) [3], Weight based Clustering Algorithm (WCA) [4] and Stable and Flexible weight based clustering algorithm (ProWBCA) [5] are mentionworthy. In HD algorithm, a node with highest degree among its 1-hop neighbors, become a clusterhead. Degree is the number of neighbors of a node. On the other hand, in LID, a node with least identification number among its neighbors, become a clusterhead. These protocols are 1-hop and completely ignore the energy and mobility

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 520–529, 2011.
© Springer-Verlag Berlin Heidelberg 2011

of nodes. On the other hand, in DCA, each node is assigned a weight (a real number ≥ 0) depending on which a node may be a clusterhead or cluster member. A node is chosen to be the clusterhead if its weight is higher than weights of its neighbors. The DCA makes an assumption that the network topology does not change during execution of the algorithm. This is not suitable for highly mobile networks.

The weight-based clustering algorithm (WCA) obtains 1-hop clusters with one clusterhead. The election of clusterhead is based on the weight of each node. The election of the clusterhead is based on weight of each node. The algorithm performs with four admissible factors for the clusterhead election and maintenance. The factors are degree difference, summation of distances, mobility and cumulative time. Although WCA has shown better performance tan all the previous algorithms, it also has some drawbacks. Weights of all nodes need to be computed before the clustering process begins. This is not suitable for a dynamic network like ad hoc networks.

ProWBCA is a 2-hop weight based clustering algorithm performs well for ad hoc networks. Weights of nodes are based on their number of 1-hop and 2-hop neighbors. However, this protocol is not power-aware or mobility-aware, which are very important from the perspective of network throughput and longevity.

In this article, we propose a fuzzy-controlled weight based clustering algorithm that considers all these aspects and constructs stable 2-hop clusters. Simulation results emphasize the effectiveness of our proposed scheme.

2 Overview of FEW-CA

Nodes in an ad hoc network may belong to various multicast groups and during cluster formation; we try to include as many members as possible of same group, into a cluster. This reduces the cost of inter-cluster communication during multicast operations. If all members of a multicast group belong to the same cluster C, then the multicast source needs to send the multicast message only to head of the cluster C only. As far as node degree is concerned, FEW- CA emphasizes on the fact that if the number of ordinary cluster members (ordinary cluster members are those that are neither cluster-heads nor gateway nodes) abruptly increase in a cluster, network throughput drops and system performance degrades [3]. On the other hand, increase in the number of gateway nodes, improve inter-cluster connectivity [1, 3]. In order to inculcate the flavor of power-awareness in FEW-CA, we have enforced another constraint that a node which doesn't have sufficient battery power cannot compete in the election of cluster-head. Also, the more power-equipped a node is, stronger will be its claim for being a cluster-head. In FEW-CA whenever a node joins a cluster, a member-join message is broadcasted by the respective cluster-head within its cluster. Similarly, when a node leaves a cluster, cluster-head broadcasts a member-lost message. If only stable clusters are constructed then the links from cluster-head to its 1-hop neighbors and links from 1-hop neighbors of the cluster-head to 2-hop neighbors of the cluster-head wont break frequently and the cost of member-join and member-lost messages will be significantly reduced.

All these observations are in the form of if-then rules which are basic unit of fuzzy function approximation. Advantages of fuzzy logic are that it is flexible, conceptually easy to understand and based on natural language. Moreover, it is tolerant of imprecise data and can model non-linear functions of arbitrary complexity. All these encouraged us to design the scheme of FEW-CA using fuzzy logic.

In FEW-CA, each node periodically broadcasts HELLO message within its radio-range and its 1-hop neighbors reply with ACK or acknowledgement message.

Attributes of these HELLO message generated by node n_i are as follows:

i) node identification number n_i
ii) radio-range R_i
iii) geographical position $(x_i(t), y_i(t))$ of n_i at time t in terms of latitude and longitude
iv) velocity $v_i(t)$ of n_i at time t
v) timestamp t
vi) cluster-head status (set to 1 if n_i is a cluster-head, otherwise it is set to 0)
vii) multicast group g of which n_i is a member ($g = 0$ if n_i does not belong to any multicast group)

After receiving the HELLO message of n_i, its neighbours reply with the ACK (acknowledgement) message. The attributes of ACK message transmitted by a neighbour n_j of n_i at time t consists of the following information:

i) source identification number n_j
ii) destination identification number n_i
iii) velocity $v_j(t)$ of n_j at time t
iv) geographical position $(x_j(t), y_j(t))$ of n_j at time t in terms of latitude and longitude
v) identification number of heads of those clusters of which n_j is a member
vi) multicast group g of which n_j is a member ($g = 0$ if n_j does not belong to any multicast group)

In case of change of cluster-head a head_change message is flooded within the respective cluster. The attributes of head_change are as follows:

i) identification number of the new cluster-head h'
ii) timestamp t

Attributes of member-join message are,

i) identification number of the new cluster member n_k
ii) timestamp t of joining the cluster
iii) status field (set to 1)

Attributes of member-leave message are,

i) identification number of the lost cluster member n_k
ii) timestamp t of leaving the cluster
iii) status field (set to 0)

3 Input Parameters of CE

Below appear the descriptions and mathematical expressions of input parameters of CE.

- ***Residual Energy Ratio***

Residual energy ratio $\alpha_i(t)$ of node n_i at time t is given by,

$$\alpha_i(t) = 1 - e_i(t) / E_i \tag{1}$$

$e_i(t)$ and E_i indicate the consumed battery power at time t and maximum or initial battery capacity of n_i, respectively. It may be noted from the formulation in 1) that $0 \leq \alpha_i(t) \leq 1$. Values close to 1 enhance capability of n_i as a cluster-head.

- ***Node Degree Influence***

The formulation of node degree influence is based on the following concepts

i) If an entire multicast group is totally contained within a cluster then it is good for network throughput and lifetime. In that case, if a source node needs to send some message to the group, then it will be sufficient to send the message to head of the cluster encapsulating that group. So, a cluster tries to include as many members as possible of same multicast groups within its territory.

ii) If most of the cluster-members and gateway nodes are at 1-hop distance, instead of 2-hop distance from the cluster-head then they will receive messages from the cluster-head at lesser delay. Therefore, it is better for a cluster if most of its members and gateways are at 1-hop distance from the cluster-head.

iii) Excessive density of nodes within a cluster greatly increases the cost of intra-cluster communication.

The formulation of node degree influence $\beta_i(t)$ of n_i at time t is given by,

$$\beta_i(t) = [\ (w\_h\ hp\_1_i(t) + hp\_2_i(t)) \times ovm_i(t)]^{1/2} \tag{2}$$

$$hp\_1_i(t) = \{\ (1/D_i(t))\ \sum_{g \in MG} \rho^1_{i,g}(t)\} \tag{3}$$

$$hp\_2_i(t) = \{\ (1/D_i(t))\ \sum_{g \in MG} \rho^2_{i,g}(t)\} \tag{4}$$

If a cluster member is at 1-hop distance from the clusterhead then minimum and maximum possible distance of the node from the clusterhead are 1 and R_{max} resulting into the average distance of $(1+ R_{max})/2$; similarly, as far as 2-hop distance of a cluster member from the clusterhead are concerned, the minimum and maximum values are $(1+ R_{min})$ and $2\ R_{max}$, resulting into the average value $\{(1+ R_{min})/2 + R_{max}\}$. It is quite practical that if a cluster member is at 1-hop delay from the clusterhead then it will receive messages from clusterhead much faster than 2-hop distant cluster members. w_h is a measure of this advantage of 1-hop neighbors of clusterhead compared to 2-hop neighbors of the same, in terms of delay. It is mathematically expressed as,

$$w\_h = \{(1+ R_{min} + 2R_{max}\} / (1+ R_{max}) \qquad (5)$$

MG is the set of various multicast groups in the network. $\rho^1_{i,g}(t)$ indicates the number of nodes belonging to the multicast group g that reside within 1-hop neighborhood of n_i at time t. Similarly, $\rho^2_{i,g}(t)$ indicates the number of nodes belonging to the multicast group g that are 2-hops away from n_i at time t. $D_i(t)$ is the number of all nodes within 2-hop neighborhood of n_i at time t.

Let N(t) be total number of nodes in the network at time t and A be the total network area. Then network density $\psi(t)$ at time t is,

$$\psi(t) = N(t) / A \qquad (6)$$

Under uniform node density, total number of nodes $\phi_i(t)$ within 2-hop neighborhood of n_i at time t is,

$$\phi_i(t) = \psi(t) \times \pi\, (R_i + R_j)^2 \qquad (7)$$

Where R_i is the radio-range of n_i and n_j is the downlink neighbor of n_i having highest radio-range among all the downlink neighbors of n_i at time t. Then neighborhood overload immunity $ovm_i(t)$ on n_i at time t is given by,

$$ovm_i(t) = \begin{cases} D_i(t) / \phi_i(t) & \text{if } D_i(t) \leq \phi_i(t) \\ \phi_i(t) / (Z + D_i(t)) & \text{otherwise} \end{cases} \qquad (8)$$

where Z is the total number of nodes in the network.

The expression in (8) is based on the fact that the cluster of n_i is not overloaded till $D_i(t) < \phi_i(t)$. So, $ovm_i(t)$ increases as $D_i(t)$ approaches $\phi_i(t)$ and acquires maximum value 1 when $D_i(t) = \phi_i(t)$. On the other hand, if $D_i(t) > \phi_i(t)$, overload immunity takes a very low fractional value which decreases with increase in $D_i(t)$.

- ***Neighbor Cluster Connectivity***

Neighbor cluster connectivity $\zeta_i(t)$ of the cluster with head n_i at time t is formulated as,

$$\zeta_i(t) = (g_i(t) / D_i(t)) \exp\, (1/(nc_i(t) +1)) \qquad (9)$$

$g_i(t)$ is the number of gateway nodes of the cluster with head n_i at time t and $nc_i(t)$ is the number of neighbor clusters of the cluster with head n_i at time t. The expression in (9) is based on the concept that it is good to have connectivity with various neighbor clusters than having multiple gateway nodes providing connectivity with same cluster.

- ***Cluster Stability***

A 2-hop cluster whose head is n_i, will be termed as stable provided its links do not break frequently. Necessary conditions for this are i) velocity of the cluster-head n_i relative to its one hop neighbors, is small ii) distance of the 1-hop neighbors of n_i from n_i is small compared to the radio-range of n_i i.e. R_i iii) for each 1-hop neighbor n_j of n_i, relative velocity of n_j with its 1-hop neighbors, is small, and iii) for each 1-hop neighbor n_j of n_i, distance of n_j from its 1-hop neighbors, is small compared to the radio-range of n_j i.e. R_j.

Among the above mentioned conditions, the first two are more important because if a cluster-head has a stable 1-hop neighborhood and unstable 2-hop neighborhood then chances are high that the cluster will not survive as a two-hop cluster but will survive as a one hop cluster. But, on the other hand, if 1-hop neighborhood of a cluster is unstable then the cluster won't survive irrespective of the stability of its 2-hop neighborhood. So, in the expression of cluster stability, stability of links from cluster-head to its 1-hop neighbors will dominate compared to stability of the links from 1-hop neighbors of cluster-head to 2-hop neighbors of the cluster-head. Hence cluster stability $cs_i(t)$ of a cluster with head n_i at time t is given by,

$$cs_i(t) = \{ \prod_{n_j \in B_i(t)} w_{ij}(t) \ \ (\sum_{n_k \in B_j(t)} w_{jk}(t)) / (|B_j(t)|+1)\} \exp(-2\,|B_i(t)|) \tag{10}$$

where $w_{ij}(t) = 1 - (1-1/(|v_i(t) - v_j(t)|+1)) \exp(dist_{ij}(t)/R_i)$ (11)

For any node n_i, $v_i(t)$ specifies its velocity at time t; $B_i(t)$ is the set of 1-hop neighbors of n_i and $dist_{ij}(t)$ is the distance between n_i and n_j at time t. The constant 1 is added in the denominator in expression (10) to avoid zero value in the denominator in the situation when 1-hop neighborhood of a 1-hop neighbor of cluster-head is empty. Similarly, 1 is added in expression (11) to avoid zero value in the denominator when $(v_i(t) - v_j(t))$.

It is evident from (10) that for any node n_i, $cs_i(t)$ ranges between 0 and 1. Stability of a cluster with head n_i increases as $cs_i(t)$ approaches 1.

4 Rule Bases of CE

The division of input parameters of CE into crisp ranges and the corresponding fuzzy variables are shown in table 1.

Table 1. Division of parameters of CE into crisp ranges and corresponding fuzzy variables

Range division of residual energy ratio	Range division of other parameters of CE	Fuzzy variables
0-0.40	0-0.25	a1
0.40-0.60	0.25-0.5	a2
0.60-0.80	0.5-0.75	a3
0.80-1.00	0.75-1.00	a4

According to the study of discharge curve of batteries heavily used in ad hoc networks, at least 40% (fuzzy variable a1 represents the range 0-0.40) of total charge is required to remain in operable condition; 40%-60% (fuzzy variable a2) of the same is satisfactory, 60%-80% (fuzzy variable a3) is good and the next higher range (i.e. 80%-100% or fuzzy variable a4) is more then sufficient from the perspective of remaining energy. Range division of residual energy ratio follows this concept. Ranges of all other parameters are uniformly divided between 0 and 1; the range 0-0.25 is denoted as a1, 0.25-0.5 is denoted as a2, 0.5-0.75 as a3 and 0.75-1.0 as a4.

It may please be noted that the parameters residual energy ratio and cluster stability are extremely important from the perspective of existence of the cluster while the other parameters focus on efficiency of the cluster. Hence, residual energy ratio α and cluster stability cs dominate other input parameters of CE. Table 2 shows the combination of residual energy ratio and cluster stability. Both are given equal weight. The temporary output t1 of table 2 is combined with node degree influence in table 3 generating second temporary output t2. Table 4 shows the combination of t2 and neighbor cluster connectivity generating output node-weight of CE. In tables 3 and 4, the output of previous table dominates the new output. The reason is that, t1 is the combination of two parameters both of which are extremely important from the point of view of existence of a cluster, and the effect propagates from t1 to t2.

Table 2. Fuzzy Combination of α and cs producing output t1

α→ cs↓	a1	a2	a3	a4
a1	a1	a1	a1	a1
a2	a1	a2	a2	a2
a3	a1	a2	a3	a3
a4	a1	a2	a3	a4

Table 3. Fuzzy Combination of t1 and β producing temporary output t2

t1→ β↓	a1	a2	a3	a4
a1	a1	a2	a3	a3
a2	a1	a2	a3	a3
a3	a1	a2	a3	a4
a4	a2	a3	a3	a4

Table 4. Fuzzy Combination of t2 and ζ producing output node-weight of CE

t1→ ζ↓	a1	a2	a3	a4
a1	a1	a2	a3	a3
a2	a1	a2	a3	a3
a3	a1	a2	a3	a4
a4	a1	a2	a3	a4

5 Simulation Results

Simulation environment appears in table 5. We compare the performance of FEW-CA with the protocols LID, HD and ProWBCA. The performance metrics are message cost, packet delivery ratio, number of clusters and rate of change of cluster by members. ns-2 [6] simulator has been used for the purpose of simulation.

Table 5. Simulation Parameters

Parameter	**Value**
Network Area	900×900 m^2 in first ten runs, 2000×1000 m^2 in nest ten runs, 900×3000 m^2 in last ten runs
Transmission Range	10 – 50 m in first ten runs, 30 – 100 m in next ten runs, 10 – 100 m in last ten runs
Interval between consecutive HELLO messages	20 seconds for first ten simulation runs, 30 seconds for next ten and 45 seconds for last ten simulation runs
Number of nodes	100 – 500
MAC layer	IEEE 802.11g
Traffic type	Constant bit rate (128 kbps/second)
Maximum number of retries before an acknowledgement is obtained	4
Packet Size	64 bytes in first ten runs, 128 bytes in next ten runs, 256 bytes in last ten runs (in different simulation runs)
Bandwidth	1- 4 Mbps in first ten runs, 2 – 7 Mbps in first ten runs, 1-10 Mbps in last ten runs
Mobility model	Random waypoint mobility model in first 10 runs, Random walk mobility model in subsequent 10 runs and Gaussian model in last 10 runs
Simulation Time	1000 s for each run
Node velocity	5-25 m/s

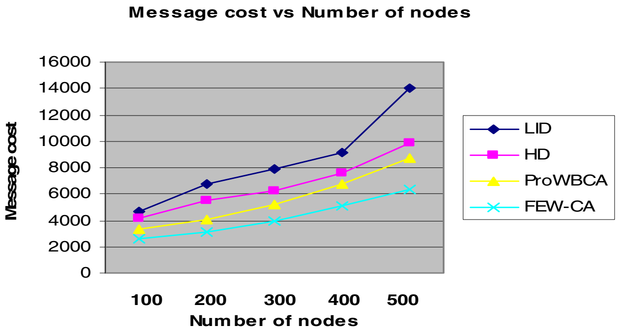


Fig. 1. Graphical illustrations of message cost vs number of nodes

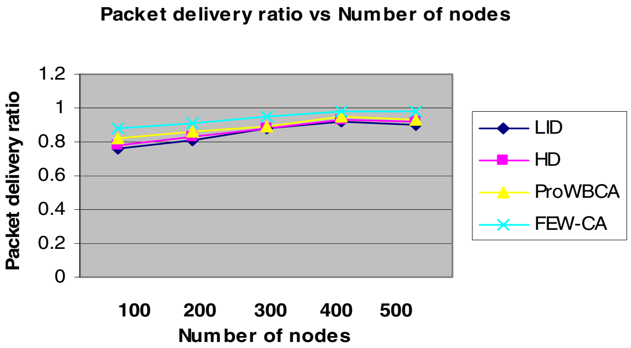


Fig. 2. Graphical illustrations of packet delivery ratio vs number of nodes

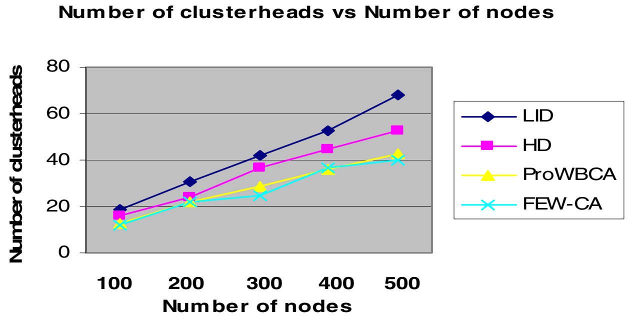


Fig. 3. Graphical illustrations of number of clusterheads vs number of nodes

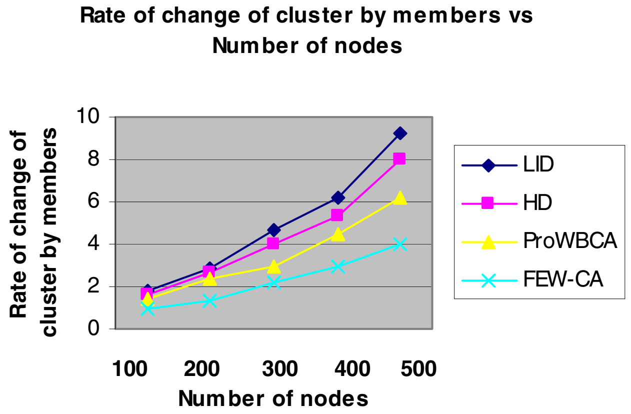


Fig. 4. Graphical illustrations of rate of change of cluster vs number of nodes

Figure 1 shows the comparison of the clustering schemes w.r.t. message cost. Unlike LID, HD and ProWBCA, FEW-CA considers stability of links and prefers only stable links for communication. Hence, the possibility of link breakage and as a result, the need for injecting link repair messages in the network decrease in FEW-CA. The cost of head_change, member_join and member_leave messages is also much less in our proposed algorithm, because the clusterhead has good connectivity with the members. All these significantly contribute in decreasing message cost in FEW-CA than its competitors. Less message cost generates less signal collision improving the packet delivery ratio as evident from figure 2. Figure 3 compares the number of clusterheads of the mentioned algorithms. Since LID and HD construct only single hop clusters, number of clusterheads in them is much higher than FEW-CA and ProWBCA. Please note here that the number of clusterheads in FEW-CA and ProWBCA don't differ much as both are 2-hop clustering algorithm. Figure 4 illustrates the rate of change of cluster by members of the algorithms LID, HD, ProWBCA and FEW-CA. In FEW-CA, the clusterheads maintain stable connectivity with its members. As a result, possibility of leaving a cluster by a cluster member is much less in FEW-CA.

6 Conclusion

This paper has presented a fuzzy-controlled weight-based clustering algorithm FEW-CA for mobile ad hoc networks. Compared to conventional clustering algorithms, this proposed 2-hop clustering algorithm is more stable w.r.t. topology changes. The algorithm is power-aware and significantly improves network throughput at much lesser cost.

References

1. Agarwal, R., Motwani, M.: Survey of Clustering Algorithm for MANETS. International Journal on Computer Sc. and Engineering 1(2), 98–104 (2009)
2. Correa, B., Ospina, L., Hincape, R.C.: Survey of clustering techniques for mobile ad hoc networks, pp. 145–161 (2007) ISSN 0120-6230
3. Yu, J.Y., Chong, P.H.J.: A survey of clustering schemes for mobile ad hoc networks. IEEE Communications Surveys and Tutorials 7(1) (2005)
4. Yang, W.D., Zhang, G.Z.: A weight-based clustering algorithm for mobile ad hoc networks. In: IEEE Proc. ICWMC 2007 (2007)
5. Pandi Selvam, R., et al.: Stable and flexible weight-based clustering algorithm in mobile ad hoc networks. International Journal on Computer Science and Information Technologies 2(2), 824–828 (2011)
6. `www.isi.edu/nsnam/ns`

Automatic Extractive Text Summarization Based on Fuzzy Logic: A Sentence Oriented Approach

M. Esther Hannah[1], T.V. Geetha[2], and Saswati Mukherjee[3]

[1] Anna University, Chennai, India
`hanmoses@yahoo.com`
[2] College of Engineering, Anna University, Chennai, India
`tv_g@hotmail.com`
[3] College of Engineering, Anna University, Chennai, India
`msaswati@yahoo.com`

Abstract. The work presents a method to perform automatic summarization of the text through sentence scoring. We propose a method which utilizes the facilities of fuzzy inference system for the purpose of scoring. Preprocessing of the text is done since this technique has its own importance enabling us to filter high quality text. A thorough review of the concepts of summarization enabled us to make use of a group of features which are very appropriate for automatic text summarization. Experimental results obtained by the proposed system on DUC 2002 data reveal that it works to the optimality with respect to other existing methods, and hence is a concrete solution to text summarization.

Index Terms: Sentence scoring, Feature extraction, Fuzzy inference, Summarization.

1 Introduction

The increasing availability of online information has necessitated intensive research in the area of automatic text summarization within the Natural Language Processing Community. Extensive use of internet is one of the main reasons why automatic text summarization draws substantial interest [9]. It provides a solution to the information overload problem which is continually encountered in the digital era. Research on automatic text summarization has a very long history, which dates back to at least 40 years, to the first system built at IBM in 1958 [13]. Many innovative approaches were explored which includes statistical and information-centric approaches, linguistic approaches and their combinations.

Though there are a number of statistical and linguistics approaches to extractive text summarization systems, most of these involve tasks that are tedious to perform, time consuming, may not be accurate and inappropriate for a given task. In this work, we propose a method that uses a unique set of all important features, makes use of sound preprocessing of text and provides better precision in sentence scoring, thereby obtaining better and more appropriate summaries. We propose a robust technique for

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 530–538, 2011.
© Springer-Verlag Berlin Heidelberg 2011

extractive text summarization that optimizes feature extraction by making use of a balanced set of features. The set of features so selected not only depends on a given sentence but is based on the overall centrality of the sentence with respect to the entire document.

Although genetic algorithms [26, 27] and neural networks [28] can perform just as well as fuzzy logic in applications relating to learning and retrieving natural language, fuzzy logic [29] has the advantage that it is built atop the structures of qualitative description used in everyday language. Since the basis for fuzzy logic is the basis for human communication, it makes it easier to automate tasks that can be successfully performed by humans [30]. In this paper we provide a fuzzy based approach to extractive text summarization where fuzzy rules are used to generate the score for every sentence based on its sentence features. The scoring enabled us to classify every sentence in the text into one of the three categories namely important, unimportant and average based on the importance of the sentence.

The rest of the paper is organized as follows: Section II brings out the existing work done on text summarizations focusing on the contributions that builds the research in this subfield of NLP. Section III progresses to discuss the proposed work under various subsections namely, preprocessing, feature extraction and fuzzy inference system. Section IV discusses the evaluation method we have used and the results are provided. Section V concludes the paper with providing scope for future work.

2 Related Works

With the use of machine learning techniques with NLP, a number of approaches that employed statistical techniques to produce document extracts came into picture. Various researchers used the feature extraction process and extracted features such as ‘frequency of term’ [1], sentence position [31], etc., and some used a linear combination of features such as cue words, keywords, title or heading and sentence location [2][7]. *Kupiec et al*, used human generated abstracts as training corpus, from which he produced extracts. The feature set included sentence length, fixed phrases, sentence position in paragraph, thematic words and uppercase words [7]. Feature extraction techniques were used to locate the important sentences in the text. *Conroy & O’leary* in 2001 modeled the problem of extracting a sentence from a document using Hidden Markov Model (HMM) [20]. In 2004, *Khosrow Kaikhah et al* proposed a new technique for summarizing news articles using a neural network that is trained to learn characteristics of sentences that should be included in the summary of the article [21]. *S.P. Yong et al w*orked on developing an automatic text summarization system combining both statistical approach and neural network to summarize documents [22].

Hsun-Hui Huang, Yau-Hwang Kuo, Horng-Chang Yang at al in 2006, proposed to extract key sentences of a document as its summary by estimating the relevance of sentences through the use of fuzzy-rough sets [23]. Ladda Suanmali et al [19] in 2009 developed a system that generates extractive summaries based on a set of features that represent the sentences in a text. The Fuzzy approach was applied to enable efficient categorization of sentences that form the summary. Our system makes use of a unique set of features which we feel is important in weighing the importance of a sentence. This set is obtained by combining features such as numerical data & pronoun feature with the ones used in the above method of text summarization. We build a summarization model that exploits the benefits of fuzzy inference.

3 Proposed Works

The proposed method delivers a technique to perform automatic summarization of the text through sentence scoring based on fuzzy inference system. First, a text, as an input, is preprocessed and high quality text is obtained. Several characteristics of each sentence such as title feature, sentence length, term weight, etc. are taken into consideration and the corresponding feature values are obtained. These feature values are used in the fuzzy inference system to generate score for every sentence based on which overall ranking of each of the sentences is done. The importance of a sentence and its presence or absence in the generated summary is based on these features as a whole, rather than on any specific feature. We have used a well-defined set of fuzzy rules in the fuzzy engine such that the rules are triggered based on the cumulative weight of all the features for every sentence. Thus the proposed method uses a sentence-oriented approach, rather than a feature-oriented one.

A set of fuzzy rules are defined, which forms the base for sentence scoring. *Figure 1* shows the proposed system architecture. The proposed work is divided into modules, namely preprocessing, feature extraction and sentence scoring using fuzzy inference, which are discussed in the following subsections:

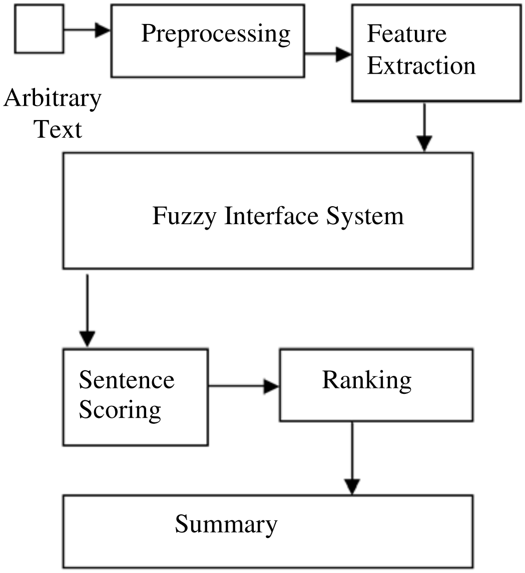


Fig. 1. System Architecture

3.1 Preprocessing

Preprocessing is done as a means of cleaning the document by removing words that do not contain any information that uniquely identifies a sentence. This module filters the noise away. Preprocessing includes Sentence segmentation, Tokenization, Stop word removal and Word stemming respectively.

3.2 Feature Extraction

For any task of text mining, features play an important role. Features are attributes that attempt to represent data used for the task. We focus on seven features for each sentence, viz. sentence length, title resemblance, thematic words, numerical data, proper noun, term frequency and sentence to sentence similarity. The rest of this section discusses these features and methods of extraction.

a) Title feature
The number of title words in the sentence contributes to title feature. Titles contain group of words that give important clues about the subjects contained in the document. Therefore if a sentence has higher intersection with the title words, the sentence is more important than others.

b) Sentence length
The number of words in sentence gives good idea about the importance of the sentence. This feature is very useful to filter out short sentences such as datelines and author names commonly found in articles. The short sentences are not expected to belong to the summary.

c) Term weight
The term weight feature score is obtained by calculating the average of the TF-ISF (Term frequency, Inverse sentence frequency). The frequency of term occurrences within a document has often been used for calculating the importance of sentence. Inverse term frequency helps to identify important sentences that represent the document.

d) Sentence to sentence similarity
Similarity between sentences is calculated as follows: for a sentence s, the similarity between s and all other sentences is computed by the cosine similarity measure. The score of this feature for a sentence is obtained by computing the ratio of the summation of sentence similarity of a sentence s with each of the other sentences over the maximum value sentence similarity.

e) Proper noun
The proper noun feature gives the score based on the number of proper nouns present in a sentence, or presence of named entity in the sentence. Usually sentences that contain proper nouns are considered to be important and these should be included in the document summary.

f) Thematic word

The number of thematic word in sentence is an important feature because terms that occur frequently in a document are probably related to topic. Thematic words are words that capture main topics discussed in a given document. We used the top 10 most frequent content word for consideration as thematic.

g) Numerical data

This feature gives a score for the number of numerical data in a sentence. The contribution made by this feature to the weight given to a sentence is significant since a sentence that contains numerical data essentially contains important information.

Each of these seven features is given a value between '0' and '1'. We have used 30 percent compression ratio for the generated summary.

3.3 Sentence Scoring Using Fuzzy Inference

The feature values extracted from the features discussed in the previous section are given to the fuzzy inference system to identify the most important sentences of the given document and these sentences are scored accordingly.

3.3.1 Fuzzy Inference System

A fuzzy inference system is a control system based on fuzzy logic which is a mathematical system that analyzes analog input values in terms of logical variables that take on continuous values between 0 and 1 in contrast to classical or digital logic, which operates on discrete values of either 0 or 1 (true or false) [15]. There are various membership function types used in fuzzy logic such as: sigmoid, Gaussian, trapezoidal (trapmf), triangular (trimf), etc. We have used the trapezoidal membership function for fuzzification since it has been used in various systems that make use of fuzzy model [25]. It is also simple and widely used and hence we have used it for the proposed fuzzy inference system.

Input given to the fuzzy inference system is fuzzified, and the fuzzified value is later defuzzified to obtain an output that aids to know the importance of every sentence.First, the features extracted are used as input to the fuzzy inference system. The score of the seven features lies between 0 and 1, and they are manually categorized as *very low(vl), low(l), average(a), high(h), very high(vh)* respectively. Thus if the score of sentence length feature is 0.8, then we classify that feature as *very high*. These input variables are fuzzified into three fuzzy sets namely *less*, *average* and *more* using the trapezoidal membership function.

Fuzzy inference engine is constructed making use of a set of fuzzy rules and this enables us to categorize the sentence into one of the output variables. The 'if' part of a rule is called the rule antecedent, and is a description of a process state in terms of a logical combination of atomic fuzzy propositions. The 'then' part of the rule is called the rule consequent and is a description of the control output in terms of logical combinations of fuzzy propositions. A sample of the IF-THEN rules is given in equation (8).

IF (vl is less) AND (l is less) AND (avg is less) AND (h is more) AND (vh is more) THEN (sentence is important) (8)

The reverse of fuzzification is called defuzzification. Based on the importance of a sentence, the fuzzy system defuzzifies every sentence into one of the three outputs variables namely *unimportant, average, and important*. This helps us to identify whether a sentence should be present in the summary or not. Such a categorization aids us in summarizing a text document.

Once the sentences are categorized, we are able to generate an information-rich summary by taking into account only sentences that are ranked as *'important'*. The top '*important'* sentences chronologically arranged are used to form the summary, based on the compression ratio that is required. In cases where the needed summary size is not achieved, '*average*' important sentences are also included in the summary. However "*unimportant*" are not considered even in such cases.

4 Evaluation and Results

The TIPSTER program with its two main evaluation style conference series TREC & Document Understanding Conference-DUC (now called as Text Analysis Conference-TAC) have shaped the scientific community in terms of performance, research paradigm and approaches. We used 55 documents from DUC2002 for testing the system. Model summaries provided by DUC2002 were used to evaluate our system. Precision, recall, and F-measure are used in the evaluation which can be calculated as follows using equations (9):

$$\text{Recall} = \frac{(S_{ref} \cap S_{sys})}{S_{sys}} \quad \text{Precision}=\frac{(S_{ref} \cap S_{sys})}{S_{ref}} \quad \text{F-score} =\frac{(\alpha+1)*\text{Recall}*\text{ Precision}}{\text{Recall} + (\alpha * \text{Precision})} \tag{9}$$

where S_{ref} and S_{sys} denote the number of segments appeared in the reference summary and in the system generated summary, respectively. For F-measure, the experiment uses F1 (i.e., the value of α is 1).

A set of metrics called *'Recall Oriented Understudy for Gisting Evaluation (ROUGE)'* [12], which has become the standards of automatic evaluation of summaries. ROUGE calculation is based on various statistical metrics by counting overlapping units such as n-grams, word sequences, and word pairs between systems which generate summaries correlating with those extracted by human evaluations. ROUGE-N is an N-gram recall between an automatic summary and a set of manual summaries.

Among the different values of *N*, unigram-based ROUGE score (ROUGE-1) has been shown to agree with human judgments the most [24]. The Precision, Recall and F-score values of the compressed text are generated using ROUGE-1, which is claimed to suit extractive summarization systems better. We have the results of sentence oriented summarizer that uses fuzzy approach and the results are promising. *Table 1* shows the results of the proposed system.

Table 1. Recall, Precision and F-Score values of sentence oriented system

F-Score	Sentence Oriented Approach (No of documents)		
Summarizer Value	Recall	Precision	F-score
<0.3	0	0	0
0.3 - 0.4	8	1	1
0.4 - 0.5	19	42	45
0.5 - 0.6	25	9	4
>=0.6	3	3	5

Table 2. R, P and F values of proposed system with other methods

ROUGE-1	Sentence Oriented Approach	Feature oriented Approach	Baseline of DUC2002	MS word 2007
Recall	0.4918	0.4666	0.44147	0.39306
Precision	0.4734	0.47589	0.44766	0.46718
F-score	0.4824	0.47019	0.44132	0.42691

The Fuzzy based sentence oriented summarizer system produced an average Recall of 0.4918, an average Precision of 0.4734 and an average F-score of 0.4824. Table-2 compares the average Recall, Precision and F-score results for ROUGE-1 produced by our sentence-oriented approach with the following: (i) Feature oriented approach using fuzzy (ii) Baseline of DUC 2002 (iii) MS-Word 2007.

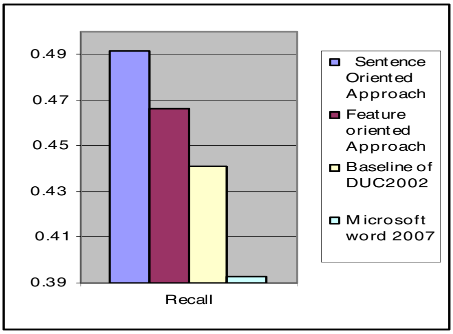


Fig. 2. Comparison of Recall

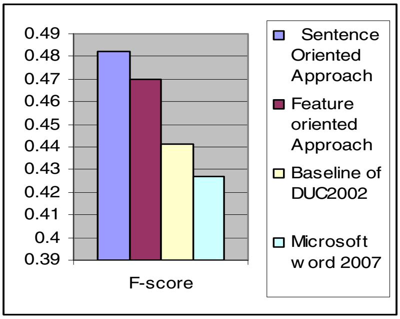


Fig. 3. Comparison of F-score

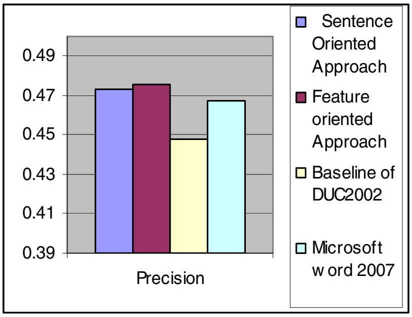


Fig. 4. Comparison of Precision

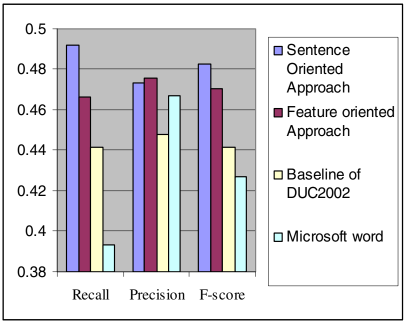


Fig. 5. Comparison of Recall, Precision, F-score value

The above results clearly show that the proposed system promises to be better than the existing ones for extractive text summarization. The Graphical representation of the results is as follows:

5 Conclusion

The proposed method provides a fuzzy based approach to text summarization, where the system selects the right set of candidates for the summary. It also can improve the quality of summarization since it extracts sentences using the scores derived from fuzzy inference system. Experimental results show that the proposed method outperforms other automatic text summarization methods. Testing this scoring methodology for other subtasks of text mining is a possible future work. We believe that by using our approach for sentence scoring and by building solutions for summarization that adhere to our proposal, redundant work that typically characterizes summarization can be avoided.

References

1. Luhn, H.P.: The Automatic Creation Of Literature Abstract. IBM Journal of Research and Development 2, 159–165 (1958)
2. Edmundson, H.P.: New Methods in Automatic extracting. Journal of the Association for Computing Machinery 16(2), 264–285 (1969)
3. Salton, G., Buckley, C.: Term_Weighting Approaches in Automatic Text Retrieval. Information Processing and Management 24, 513–523 (1997)
4. Salton, G.: Automatic Text Processing: The, Analysis, Transformation, and Retrieval of Information by Computer. Addison-Wesley Publishing Company (1989)
5. Fattah, M.A., Ren, F.: Automatic Text Summarization. Proceedings of World Academy of Science, Engineering and Technology 27, 192–195 (2008)
6. Lin, C.Y.: Training a Selection Function for Extraction. In: Proceedings of the Eighth International Conference on Information and Knowledge Management, Kansas City, Missouri, United States, pp. 55–62 (1999)
7. Kupiec, J., Pedersen, J., Chen, F.: A Trainable Document Summarizer. In: Proceedings of the Eighteenth Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR), Seattle, WA, pp. 68–73 (1995)
8. Kulkarni, A.D., Cavanaugh, C.D.: Fuzzy Neural Network Models for Classification. Applied Intelligence 12, 207–215 (2000)
9. Mani, I., Maybury, M.: Advances in automatic text summarization. MIT Press (1999)
10. Radev, D.R., Hovy, E., McKeown, K.: Introduction to the special issue on summarization, vol. 28(4), pp. 399–408 (2002)
11. Mani, I., Klein, G., House, D., Hirsctman, L., Firmin, T., Sundheim, B.: SUMMAC: a text summarization evaluation. Natural Language Engineering 8(1), 43–68 (2002)
12. Lin, C.-Y.: Rouge: A package for automatic evaluation of summaries. In: Proceedings of the ACL 2004 Workshop, pp. 74–81 (2004)
13. Fischer, S., Roark: Query-focused summarization by supervised sentences ranking and skewed word distribution. In: Proceedings of DUC (2006)

14. Freund, Y., Scaphire, R.E.: Experiments with a new boosting algorithm. In: Proceedings of the Thirteenth International Conference on Machine Learning (1996)
15. Pal, S.K., Mithra, P.: Case generation using rough sets with fuzzy representations. IEEE Transactions on Knowledge and Data Engineering 16(3) (March 2004)
16. Dubois, D., Prade, H.: Putting rough sets and fuzzy sets together. Intelligent Decision Support, 203–232 (1992)
17. Svore, K., Vanderwende, L., Burges, C.: Enhancing single document summarization by combining RankNet and third-party sources. In: Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 448–457. Association for Computational Linguistics, Prague (2007)
18. Radev, D.R., Hovy, E., McKeown, K.: Introduction to the special issue on summarization. Computational Linguistics 28(4), 399–408 (2002)
19. Suanmali, L., Binwahlan, M.S., Salim, N.: Sentence Features Fusion for Text Summarization using Fuzzy Logic. In: Proceedings of Ninth International Conference for Text Summarization Using Fuzzy Logic, pp. 142–146 (2009)
20. Conroy, J.M., O'leary, D.P.: Text summarization via hidden markov Models. In: Proceedings of SIGIR 2001, pp. 406–407 (2001)
21. Kaikhah, K.: Text Summarization Using Neural Networks. Faculty Publications, Texas State University (2004)
22. Yong, S.P., Abidin, A.I.Z., Chen, Y.Y.: A Neural Based Text Summarization System. In: Proceedings of the 6th International Conference of Data Mining (2005)
23. Huang, H.-H., Kuo, Y.-H., Yang, H.-C.: Fuzzy-Rough Set Aided Sentence Extraction Summarization. In: Proceedings of the First International Conference on Innovative Computing, Information and Control (2006)
24. Lin, C.Y., Hovy, E.H.: The Potential and Limitation of Sentence Extraction for Summarization. In: Proceedings of the HLT/NAACL Workshop on Automatic Summarization, Edmonton, Canada (2003)
25. Rubens, N.O.: The Application of Fuzzy Logic To The Construction Of The Ranking Function of Information Retrieval Systems. Computer Modelling and New Technologies 10(1), 20–27 (2006)
26. Sakakibara, Y.: Learning context-free grammars using tabular representations. Pattern Recognition 38(9), 1372–1383 (2005)
27. Keller, B., Lutz, R.: Evolutionary induction of stochastic context free grammars. Pattern Recognition 38(9), 1393–1406 (2005)
28. Sakakibara, Y., Golea, M.: Simple recurrent networks as generalized hidden markov models with distributed representations. In: Proceedings of IEEE International Conference on Neural Networks (ICNN 1995), pp. 979–984. IEEE Computer Society Press, New York (1995)
29. Wang, X., Chaudhari, N.S.: Alignment based Similarity Measure for Grammar Learning. In: Proceedings of IEEE International Conferenceon Fuzzy Systems (Fuzz-IEEE 2006), pp. 9034–9041 (2006)
30. Kyoomarsi, F., Khosravi, H., Eslami, E., Dehkordy, P.K., Tajoddin, A.: Optimizing Text Summarization Based on Fuzzy Logic. In: The Proceedings of the Seventh IEEE/ACIS International Conference on Computer and Information Science. IEEE computer Society (2008)
31. Baxendale, P.: Machine-made index for technical literature - an experiment. IBM Journal of Research Development, 354–361 (1958)

An Improved CART Decision Tree for Datasets with Irrelevant Feature

Ali Mirza Mahmood[1], Mohammad Imran[2], Naganjaneyulu Satuluri[1],
Mrithyumjaya Rao Kuppa[3], and Vemulakonda Rajesh[4]

[1] Acharya Nagarjuna University, Guntur, Andhra Pradesh, India
[2] Rayalaseema University, Kurnool, Andhra Pradesh, India
[3] Vaagdevi College of Engineering, Warangal, Andhra Pradesh, India
[4] Pursing M.Tech, MIST, Sathupalli, Khamaman District, Andhra Pradesh, India
`alimirza.md@gmail.com`

Abstract. Data mining tasks results are usually improved by reducing the dimensionality of data. This improvement however is achieved harder in the case that data size is moderate or huge. Although numerous algorithms for accuracy improvement have been proposed, all assume that inducing a compact and highly generalized model is difficult. In order to address above said issue, we introduce Randomized Gini Index (RGI), a novel heuristic function for dimensionality reduction, particularly applicable in large scale databases. Apart from removing irrelevant attributes, our algorithm is capable of minimizing the level of noise in the data to a greater extend which is a very attractive feature for data mining problems. We extensively evaluate its performance through experiments on both artificial and real world datasets. The outcome of the study shows the suitability and viability of our approach for knowledge discovery in moderate and large datasets.

Keywords: Classification, Decision trees, Filter, Randomized gini index.

1 Introduction

In Machine Learning community, and in Data Mining works, Classification has its own importance. Classification is an important part and the research application field in the data mining [1]. With ever-growing volumes of operational data, many organizations have started to apply data-mining techniques to mine their data for novel, valuable information that can be used to support their decision making [2]. Organizations make extensive use of data mining techniques in order to define meaningful and predictable relationships between objects [3]. Decision tree learning is one of the most widely used and practical methods for inductive inference [4]. Decision trees are one of the most effective machine learning approach for extracting practical knowledge from real world datasets [5].

The main contributions of this work can be summarized as follows.

(i)We show that a fast random sampling framework can be used to enhance the generalization accuracy of the tree. (ii) It is worth to note here that the main peculiarity of this composite splitting criterion is that the resulting decision tree is

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 539–549, 2011.
© Springer-Verlag Berlin Heidelberg 2011

better in accuracy. (iii) We connect the theoretical results from state-of-the-art decision tree algorithm (CART) showing the viability of our method and also show empirical results supporting our claim.

2 Related Work

In this Section, we present some recent work on decision trees in different areas, Aviad. B [6] have proposes and evaluates a new technique to define decision tree based on cluster analysis. The results of the model were compared to results obtained by conventional decision trees. It was found that the decision rules obtained by the model are at least as good as those obtained by conventional decision trees. In some cases the model yields better results than decision trees. In addition, a new measure is developed to help fine-tune the clustering model to achieve better and more accurate results. Pei-Chann Chang [7] have applied fuzzy logic as a data mining process to generate decision trees from a stock database containing historical information. They have establishes a novel case based fuzzy decision tree model to identify the most important predicting attributes, and extract a set of fuzzy decision rules that can be used to predict the time series behavior in the future. The fuzzy decision tree generated from the stock database is then converted to fuzzy rules that can be further applied in decision-making of stock price's movement based on its current condition. Leyli Mohammad Khanli [8] have applied active rule learning is regarded for resource management in grid computing. Rule learning is very important for updating rules in active database system. But, it is also very difficult because of lacking methodology and support. Decision tree can use into rule learning to cope with the problems arisen in active semantic extraction, termination analysis of rules set and rules update. Also their aim from rule learning is learning new attributes in rules such as time, load balancing regarded to instances of real Grid environment that decision tree can provide it. Ali Mirza Mahmood [9] have proposed the use of expert knowledge in pruning decision trees for applicability in medical analysis. There has been significant research interest in decision trees in recent years. In [10] author have proposed an improved decision tree classification algorithm MAdaBoost which constructs cascade structures of more decision tree classifiers based on AdaBoost for tackling the problem of imbalanced datasets. The improved algorithm eliminates the short coming of imbalance datasets and improves the overall accuracy of cascade classifiers. In [11] author proposed improved decision tree which uses series of pruning techniques that can greatly improve construction efficiency of decision trees when using for uncertain data.

3 Components of Randomized Gini Index

In this Section, we investigate to propose a new Randomized Gini Index framework (RGI). Our randomized sampling method depends on small random subset of attributes. We assume that the subset of the training data is small, i.e. it is computationally cheap to act on such a set in a reasonable time. Also, such randomized sampling is done multiple times. We focus on a set of commonly used random sampling procedure and Filter. Next, we try to adapt and deploy them as RGI components. The next stage of RGI tries to consider both gini index and weights for splitting of attributes. The quality of solution fine-tuning, mainly, depends on the nature of the filter involved and the parameters of random sampling. The following four sub sections, detail different design alternatives for both random sampling and filter procedure search for RGI components.

3.1 Random Sampling Method

Due to the large dimensionality of the feature space, it may be difficult for a search method to search appropriate features. In order to increase the computational speed we used random sampling method [12]. Randomized Sampling (RS) is the process of generating random subset datasets from the original dataset where every feature has equal chance. In random sampling we choose a subset of m features out of the presented n features such that $m << n$. In order to cover a large portion of the features in the dataset, we repeat the selection t times. The algorithm for random sampling is given in Algorithm 1.

Algorithm 1. Random sampling RS method

```
Input: $n$ examples each with $p$ features, $K$ randomized experiments.
Output: Count vector $W$ (1xD vector) representing number of times features were selected in K randomized experiment.
Procedure:
    Select K randomized sets each of size B and denote then as
    $N\,exam_i; i = 1,2,..K$ and let $V \leftarrow 0$
    for $i = 1,2,..K$ do
        Get $N\,exam_i$ set;
        Train $Model_i$ = Filter
        Si = selected features in $Model_i$ via $N\,exam_i$
        $V \leftarrow V + \{x, x \in R^D \ |x_j = 1 if j \in S_i else x_j = 0\}$
    end
```

3.2 Filter for Attribute Selection

Considered as the earliest approaches to feature selection, filter methods discard irrelevant features, without any reference to a data mining technique, by applying independent search which is mainly based on the assessment of intrinsic attribute properties and their relationship with the data set class (i.e. Relief, Symmetrical uncertainty, Pearson correlation, etc)[13].

3.3 Creating Weighted Vectors for Attributes

After many such randomized experiments, the counts of the number of times a feature was found in those randomized experiments is summed up and normalized and denoted by V.

This count vector, denoted by V, is then inverted and used as weights for the weighted version of the heuristic algorithm; i.e. weights used in the weighted formulations are $W = 1=V$. Intuitively, if a feature is important and is found multiple times via the RS method, then the corresponding weight for the feature is less and thus it is penalized lesser, encouraging higher magnitude for the feature.

The algorithm for inducing new decision tree by using RGI is shown in Algorithm 2,

Algorithm 2. New Decision Tree (D, A, RGI)

```
Input:      D   – Data Partition
         A    – Attribute List
         RGR – Randomized Gini Index (Gain)
Output:     A Decision Tree.
Procedure:
          Create a node N
         If samples in N are of same class, C then
            return N as a leaf node and mark class C
            If A is empty then
            else
            apply Randomized-Gini Gain ( $a_i, S_w$ )
            label root node N as f(A)
            for each outcome j of f(A) do
            subtree j =New Decision Tree(Dj,A,RGI)
            connect the root node N to subtree j
            endfor
         endif
      endif
      Return N
```

3.4 Inducing Decision Trees

The RGI paradigm needs a heuristic function as its base algorithm. In this paper, we combine it with one of the most popular algorithms, CART. CART has proven to be a benchmark against which the performances of machine learning algorithms are measured. As an algorithm it is robust, accurate, fast, and, as an added bonus, it produces a comprehensible structure summarizing the knowledge it induces. We propose to integrate our weighted features in CART heuristic function. In CART, gini index is used as the heuristic function to perform splitting of the nodes at the growing phase. Gini index is an impurity-based criterion that measures the divergence between the probability distributions of the target attribute's values. The Gini index has been used in various works . The Gini is defined as in equation 1 and gini gain can be calculated by using equation 2

$$Gini(y,S)=1-\sum_{c_j\in dom(y)}\left(\frac{\sigma_{y=C_j}S}{|S|}\right)^2 \tag{1}$$

Consequently, the evaluation criteria for selecting the attribute a_i is defined as

$$Gini\,Gain(a_i,S)=Gini(y,S)-\sum_{v_{i,j}\in dom(a_i)}\frac{\sigma_{ai=v_{i,j}}S}{|S|}\times Gini\left(y,\sigma_{ai=v_{i,j}}S\right) \qquad (2)$$

The new composite heuristic function Randomized -Gini index can be obtained by using equation (3),

$$RandomizedGini\,Gain(a_i,S_W)=Gini(y,S_W)-\sum_{v_{i,j}\in dom(a_i)}\frac{\sigma_{ai=v_{i,j}}S_W}{|S_W|}\times Gini\left(y,\sigma_{ai=v_{i,j}}S_W\right) \qquad (3)$$

4 Experiments on Synthetic and Real World Datasets

We performed the implementation of our new algorithm within the Weka [14] environment on windows XP with 2Duo CPU running on 2.53 GHz PC with 2.0 GB of RAM. As we mentioned, the RGI paradigm need a base learner classification algorithm. Here, we combine it with one of the most popular algorithms, CART. In this paper, for the process of new heuristic function, we have generated random samples of 5 trails for each original dataset. In second stage, we used a filter approach, to find a set of relevant features and in last stage, we exploited CART's gini index to induce decision trees. In order to test the feasibility of the proposed heuristic function, we have carried out a number of experiments on artificially generated data set, as well as real-world data sets. We choose 40 UCI [15] datasets, which are commonly used in the supervised learning research area. The details of each data set are available in Table 2. We conducted experiments by using 10 fold cross validation for 10 runs to test the performance of various methods. The summary of instantiation of the search instances for RGI is given in Table 1.

Table 1. The Summary of instantiation of the search instances for RGR

State	RGI
Initial State	The empty set of features(0,0,0 . . . 0)
Evaluator	Filter
Learning Scheme	CART
Search algorithm	Hill-climbing or best-first search
Search termination	5

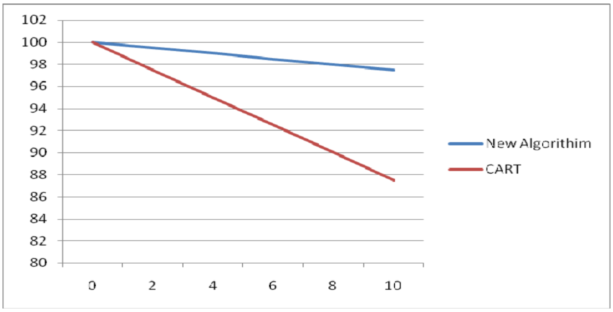


Fig. 1. Experimental results for artificial data on both algorithms

4.1 Synthetic Dataset

First, we tested our method on these artificial data generated by using XOR problem and we added artificial noise in the class labels to study its robustness. The noisy version of each training data set is generated by choosing 2% instances and changing their class labels to other incorrect labels randomly up to 10 %. The experimental results show that our new algorithm can significantly outperform CART. The problem

Table 2. The properties of the 40 UCI datasets

S.No	Dataset	Inst.	Missing values	Numeric. attributes	Nominal attributes	Classes
1.	Anneal	898	no	6	32	5
2.	Anneal.ORIG	898	yes	6	32	5
3.	Arrhythmia	452	yes	206	73	13
4.	Audiology	226	yes	0	69	24
5.	Autos	205	yes	15	10	6
6.	Balance-scale	625	no	4	0	3
7.	Breast-cancer	286	yes	0	9	2
8.	Breast-w	699	yes	9	0	2
9.	Colic-h	368	yes	7	15	2
10.	Colic-h.ORIG	368	yes	7	15	2
11.	Credit-a	690	yes	6	9	2
12.	Credit-g	1000	no	7	13	2
13.	Pima diabetes	768	no	8	0	2
14.	Ecoli	336	no	7	0	8
15.	Glass	214	no	9	0	6
16.	Heart-c	303	yes	6	7	2
17.	Heart-h	294	yes	6	7	2
18.	Heart-statlog	270	no	13	0	2
19.	Hepatitis	155	yes	6	13	12
20.	Hypothyroid	3772	yes	7	22	4
21.	Ionosphere	351	no	34	0	2
22.	Iris	150	no	4	0	3
23.	Kr-vs-kp	3196	no	0	36	2
24.	Labor	57	yes	8	8	2
25.	Letter	20000	no	16	0	26
26.	Lympho	148	no	3	15	4
27.	Mushroom	8124	yes	0	22	2
28.	Optdigits	5620	no	64	0	10
29.	Pendigits	10992	no	16	0	10
30.	Primary-tumor	339	yes	0	17	21
31.	Segment	2310	no	19	0	7
32.	Sick	3772	yes	7	22	2
33.	Sonar	208	no	60	0	2
34.	Soybean	683	yes	0	35	19
35.	Splice	3190	no	0	61	3
36.	Vehicle	846	no	18	0	4
37.	Vote	435	yes	0	16	2
38.	Vowel	990	no	10	3	11
39.	Waveform	5000	no	41	0	3
40.	Zoo	101	no	1	16	7

which we chose in artificial data is the XOR problem. We have created training and testing data contains 64 sample set of the patterns below. {0,0,0; 0,1,1; 1,0,1; 1,1,1}.After that we added ten random binary features to both training and testing data, and observed the performance of both C4.5 and our new algorithm. The experimental results conducted with artificial domains for both C4.5 and new algorithm, are summarized in Fig 1.

One can observe from the results that the accuracy of CART decreases drastically by the addition of irrelevant features. The accuracy of CART has reduced from 100 % to 87 %, where as the accuracy of new algorithm have simply changed from 100 % to 97 %, indicating that new algorithm have removed almost all the irrelevant attributes.

4.2 Real World UCI Datasets

The second type of experiments examined the ability of the algorithms in natural domains. In experiments with natural domain, we could neither vary nor measure the number of irrelevant features in these domains, we could make educated guesses about the prevalence of irrelevant features by comparing the patterns of results to those found with artificial data.

5 Experimental Results

5.1 Results on UCI Datasets

In this section, the result of the comparative analysis based on accuracy between RGR and other traditional bench mark splitting criteria's is shown. The experimental results are summarized in Table 3 to Table 6 and in Figure 2 to Figure 5. In the below tables, GR represents the gain ratio, IG represents information gain, GI represents gini index used in CART, and RGI represents Randomized gini index. The reason that these three traditional splitting criteria's are selected is that they are among the most popular splitting criteria's and many researchers used them in comparative analysis for new splitting criteria in decision trees [16], [17],[18].

Table 3 shows the detailed experimental results of the mean classification accuracy and of Gain Ratio, Information Gain, Gini Index and Randomized Gini Index method using CART as base classifier on each data set. And the mean values, overall ranks and the pairwise *t*-test results are summarized in the Table 5. From Table 3 we can see that RGI can achieve substantial improvement over GR on most data set (11 wins and 3 losses) which suggests that RGI is potentially a good heuristic function for decision trees. RGI also gain significantly improvement over IG (7 wins and 2 losses) and is comparable to GI(6 wins and 4 losses) .The overall rank of RGI on these 40 data sets is 2.22 which the smallest among all these ensemble methods. Thus , compared to RG, IG and GI which generates a decision tree with noise branch due to the presence of noise data in the datasets in the form of irrelevant features.

Table 3. Accuracy of Gain Ratio, Information Gain, Gini Index and Randomized Gini Index on the 40 UCI data sets. Comparative analysis based on accuracy between Gain Ratio, Information Gain, Gini Index and Randomized Gini index.

Dataset	**GR**	**IG**	**GI**	**RGI**
Anneal	98.38	96.89	98.25	98.50
Anneal.ORIG	90.30	90.61	91.82	93.23
Arrhythmia	65.12	65.54	71.31	71.33
Audiology	77.90	71.86	74.25	75.30
Autos	73.02	52.31	74.65	77.24
Balance-scale	78.19	77.63	78.73	72.29
Breast-cancer	71.23	66.96	70.22	94.74
Breast-w	94.30	94.01	94.74	94.74
Colic	84.91	84.43	85.37	85.64
Colic.ORIG	66.40	66.84	66.92	70.32
Credit-a	85.93	85.10	85.95	85.04
Credit-g	71.97	71.62	72.63	74.38
Pima_diabetes	73.46	74.82	74.88	74.61
Ecoli	80.62	79.42	81.03	83.17
Glass	67.32	63.57	68.91	71.91
Heart-c	77.72	74.12	78.02	78.85
Heart-h	78.29	78.44	78.59	79.16
Heart-statlog	76.89	76.57	76.73	74.89
Hepatitis	78.14	79.76	78.99	78.11
Hypothyroid	99.49	99.33	99.54	98.83
Ionosphere	88.80	89.22	88.63	88.72
Iris	94.12	94.90	94.61	93.80
Kr-vs-kp	99.18	98.70	99.14	94.02
Labor	79.38	77.81	81.24	80.13
Letter	86.23	82.16	85.12	87.24
Lympho	76.68	72.82	77.19	72.87
Mushroom	100.0	99.96	99.93	100.0
Optdigits	89.42	87.72	89.09	90.41
Pendigits	95.97	94.39	95.45	96.16
Primary-tumor	39.10	37.80	41.13	38.76
Segment	96.07	94.57	95.02	96.03
Sick	98.72	98.46	98.60	98.13
Sonar	71.12	68.77	71.26	71.80
Soybean	88.48	79.30	90.27	90.89
Splice	93.55	52.20	84.34	90.38
Vehicle	70.31	68.91	69.60	67.77
Vote	95.54	95.10	95.06	95.77
Vowel	75.59	50.52	74.39	78.23
Waveform	75.24	76.05	76.67	77.23
Zoo	93.26	40.54	40.54	40.61

Furthermore, RGI is comparable to C4.5 (GR) which is the state-of-the-art decision tree technique. Note that RGI have performed well on C4.5. Table 4 shows the detailed experimental results of the mean tree size and of Gain Ratio, Information Gain, Gini Index and Randomized Gini Index method using CART as base classifier on each data set. And the mean values, overall ranks and the pairwise *t*-test results are summarized in the Table 5.

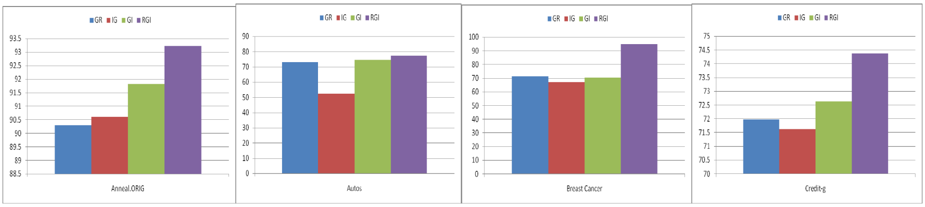


Fig. 2. Test results on accuracy between the Gain Ratio, Information Gain, Gini Index, and Randomized Gini index on anneal.orig, autos, breast-cancer, and credit-g datasets

Table 4. Tree Size of Gain Ratio, Information Gain, Gini Index and Randomized Gini Index on the 40 UCI data sets. Comparative analysis based on accuracy between Gain Ratio, Information Gain, Gini Index and Randomized Gini index.

Dataset	**GR**	**IG**	**GI**	**RGI**
Anneal	43.20	32.15	21.02	21.70
Anneal.ORIG	55.45	53.45	76.10	96.01
Arrhythmia	63.00	18.10	16.60	67.28
Audiology	41.95	28.50	35.82	35.14
Autos	51.55	36.25	43.10	48.86
Balance-scale	62.20	34.80	44.60	32.92
Breast-cancer	16.45	23.65	7.16	7.72
Breast-w	17.50	9.80	15.88	15.88
Colic	9.05	13.55	6.42	5.00
Colic.ORIG	1.00	220.7	54.57	70.32
Credit-a	32.65	19.25	4.60	9.50
Credit-g	96.80	52.95	27.40	26.74
Pima_diabetes	35.70	22.50	16.40	16.02
Ecoli	27.00	13.30	18.00	22.72
Glass	35.20	15.10	21.60	23.92
Heart-c	29.50	14.90	12.50	17.82
Heart-h	17.80	11.40	9.90	16.56
Heart-statlog	28.40	10.70	14.20	15.70
Hepatitis	11.90	5.40	7.60	83.04
Hypothyroid	21.85	16.95	16.80	13.30
Ionosphere	19.30	89.22	8.30	11.98
Iris	6.70	5.50	5.80	8.42
Kr-vs-kp	48.60	44.70	51.20	12.26
Labor	5.30	4.90	7.20	8.22
Letter	1922	984	1702	2203.7
Lympho	19.65	10.40	10.70	17.02
Mushroom	29.85	37.60	13.20	16.28
Optdigits	89.42	87.72	89.09	278.4
Pendigits	95.97	94.39	95.45	351.78
Primary-tumor	39.10	37.80	41.13	30.96
Segment	96.07	94.57	95.02	78.30
Sick	98.72	98.46	98.60	16.96
Sonar	71.12	68.77	71.26	14.66
Soybean	88.48	79.30	90.27	100.60
Splice	93.55	52.20	84.34	90.38
Vehicle	70.31	68.91	69.60	53.36
Vote	95.54	95.10	95.06	9.40
Vowel	75.59	50.52	74.39	168.70
Waveform	75.24	76.05	76.67	168.74
Zoo	93.26	40.54	40.54	1.00

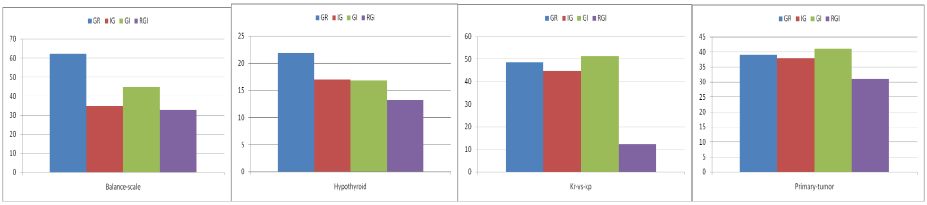


Fig. 3. Test results on tree size between the Gain Ratio, Information Gain, Gini Index, and Randomized Gini index on balance-scale, hypothyroid, kr-vs-kp and primary-tumor datasets

From Table 4 we can see that RGI can achieve substantial improvement over GR on most data set (11 wins and 3 losses) which suggests that RGR is potentially a good heuristic function for decision trees. RGI also gain significantly improvement over IG (7 wins and 2 losses) and is comparable to GI(6 wins and 4 losses). From Table 3, we can also see that RGI is significantly better than remaining benchmark algorithms.

Table 5. Win-Tie-Loss (*w/t/l*) comparisons between Randomized Gini Index against other algorithms using pairwise *t*-tests at 95% significance level, respectively

Results	**Systems**	**Wins**	**Ties**	**Losses**
Accuracy	RGR vs. GR	9	14	17
	RGR vs. IG	19	12	9
	RGR vs. GI	13	13	14
Tree Size	RGR vs. GR	33	2	5
	RGR vs. IG	19	3	17
	RGR vs. GI	19	2	18

6 Conclusion and Future Work

In this article, we propose a frame work to generate decision trees with better accuracy. Our algorithm is also effective to solve the problems discussed in gini index. We performed a series of primary and comparative experiments on 40 real-world data sets, and our method obtained encouraging results. The applications of this RGI in real-world learning tasks, especially medical data analysis tasks, will be fruitful.

There are many problems for future research. First, Analysis must be performed to ensure that the heuristic function still have near-optimal sample complexity. Secondly, we also hope to forge a stronger link between our studies of natural and artificial domain for our pruning technique.

References

[1] Zhao, H., Sinha, A.P.: An Efficient Algorithm for Generating Generalized Decision Forests. IEEE Transactions on Systems, Man, and Cybernetics —Part A: Systems and Humans 35(5), 287–299 (2005)

[2] Hu, J., Deng, J., Sui, M.: A New Approach for Decision Tree Based on Principal Component Analysis. In: Proceedings of Conference on Computational Intelligence and Software Engineering, pp. 1–4 (2009)

[3] Liu, D., Lai, C., Lee, W.: A Hybrid of Sequential Rules and Collaborative Filtering for Product Recommendation. Information Sciences 179(20), 3505–3519 (2009)

[4] Mitchell, T.M.: Machine Learning. McGraw Hill, New York (1997)

[5] Mahmood, A.M., Kuppa, M.R., Reddi, K.K.: A new decision tree induction using composite splitting criterion. Journal of Applied Computer Science & Mathematics 9(4), 69–74 (2010)

[6] Aviad, B., Roy, G.: Classification by Clustering Decision Tree-like Classifier based on Adjusted Clusters. Expert Systems with Applications (2011), doi:10.1016/j.eswa.2011.01.001

[7] Chang, P.-C., Fan, C.-Y., Lin, J.-L.: Trend discovery in financial time series data using a case based fuzzy decision tree. Expert Systems with Applications 38, 6070–6080 (2011)

[8] Khanli, L.M., Mahan, F., Isazadeh, A.: Active rule learning using decision tree for resource management in grid computing. In: Future Generation Computer Systems (2011), doi:10.1016/j.future.2010.12.016

[9] Mahmood, A.M., Kuppa, M.R.: A novel pruning approach using expert knowledge for data-specific pruning. Engineering with Computers (2011), doi:10.1007/s00366-011-0214-1

[10] Wang, Y.: The Cascade Decision-tree Improvement Algorithm Based on Unbalanced Data Set. In: Proceedings of 2010 International Conference on Communications and Mobile Computing, pp. 284–288 (2010)

[11] Tsang, S., Kao, B., Yip, K.Y., Ho, W.-S., Lee, S.D.: Decision Trees for Uncertain Data. IEEE Transactions on Knowledge and Data Engineering 23(1), 64–78 (2011)

[12] Politos, D.N., Romano, J.P., Wolf, M.: Subsampling. Springer, Heidelberg (1999)

[13] Liu, H., Motoda, H.: Computational methods of feature selection. Chapman and Hall/CRC Editions (2008)

[14] Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)

[15] Asuncion, A., Newman, D.: UCI machine learning repository (2007), `http://www.ics.uci.edu/~mlearn/MLRepository.html`

[16] Li, N., Zhao, L., Chen, A.-X., Meng, Q.-W., Zhang, G.-F.: A New Heuristic of the Decision Tree Induction. In: Proceedings of the Eighth International Conference on Machine Learning and Cybernetics, Baoding, pp. 1659–1664 (2009)

[17] Qi, C.: A New Partition Criterion for fuzzy Decision Tree Algorithm. In: Proceedings of Workshop on IntelligentInformation Technology Application, pp. 43–46 (2007)

[18] Mahmood, A.M., Kuppa, M.R.: Early Detection of Clinical Parameters in Heart Disease Using Improved Decision Tree Algorithm. In: Proceedings of IEEE 2nd Vaagdevi International Conference on Information Technology for Real World Problems (VCON 2010), Warangal, India, pp. 24–29 (2010)

Fuzzy Rough Set Approach Based Classifier

Alpna Singh, Aruna Tiwari, and Sujata Naegi

Department of Computer Engineering, SGSITS Indore India
alpna.singh87@gmail.com, atiwari@sgsits.ac.in, mrsnegi@yahoo.com

Abstract. In this paper a fuzzy rough set approach based classifier is proposed. To design this classifier, fuzzy approximation operator proposed by Zhao, Tsang and Chen 2010 [1] has been modified and new rules are proposed for classification. The fuzzy rough set based classification rules are used to predict decision class of new objects with unknown class. To extract these rules, first we build the equivalence classes, calculate the lower approximation value and then make use of a constant degree to reduce redundant attribute values. By using this concept, we design the discernibility vector, attribute value core of every object, to develop an attribute value reduction algorithm. These rules are applied on benchmark of dataset and classification accuracy is measured. Experimental results have been carried out and it shows that number of rules, training time of classifier is reduced and classification accuracy is improved on some dataset.

Keywords: Fuzzy-rough set, discernibility vector, classifier, classification rules, fuzzy rough approximation operator.

1 Introduction

The most successful approach to deal with uncertainty and manipulate imperfect knowledge is fuzzy set theory and rough set theory. In fuzzy set theory, fuzzy set is a set that allows its member to have different degree of membership called membership function. Membership function defines whether this element belongs to a set or not. As a mathematical approach, rough set theory is used to handle imprecision, vagueness, and uncertainty. Rough set theory expresses vagueness not by means of membership function, but by employing a boundary region of a set. In set the values of attributes could be both of symbolic and real-valued. So there is a need of some methods which have the capability of utilizing set approximations, attributes reduction and real values attributed datasets. This could be possible by combining fuzzy sets and rough sets. This combination of concept is termed as fuzzy rough sets (FRS) [2]. FRS is mainly used for reducing the complexity of the set and also build classifier on classification problems. Fuzzy rough set approach based classifier is designed to classify the decision class of new objects with unknown class, for purpose of decision making. Fuzzy rough set approach based classifier perform well on the dataset containing real number values. It reduces the complexity of the set and shows improvement in terms of classification accuracy on some dataset.

The outline of this paper is as follows. Section-2 describes the concepts of fuzzy rough set approach. Section-3 We calculates the lower approximation operator and

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 550–558, 2011.
© Springer-Verlag Berlin Heidelberg 2011

constant degree to reduce superfluous attribute using strictly mathematical reasoning. Section-4 We construct discernibility vector of each object, attribute value core of each object and proposed attribute value reduction algorithm. Section-5 presents how we extracted the rules to classify the unknown objects and experimental results are presented. Finally we conclude this paper in Section-6.

2 Fuzzy Rough Set Approach Concept

The basic concept of FRS was first proposed by Dubois Parade. FRS is composed of two parts one was knowledge representation (rough approximation of fuzzy sets), while other was knowledge discovery (attribute value reduction and extracting some rules). The Rough approximation of fuzzy set is done by using two approaches [5], [6]. First, is the axiomatic approach, various classes of fuzzy rough approximation operators are characterized by different sets of axioms. The approximation operator used in FRS focused on studying the mathematical structure of FRS such as algebraic and topological structure. Second, is constructive approach in which a pair of lower and upper FRS operators is defined. As per literature[5][6] let U be a nonempty set with finite objects, R a fuzzy binary relation on U, A(u) membership degree. The fuzzy lower and upper approximations operator are presented using membership function. The membership function representation of fuzzy approximation operator is generalized by introducing threshold. α is a threshold α ε [0,1]. Fuzzy Lower approximation operator [1] are defined as follows :

$$\underline{R_{s\propto}}A(x) = \inf_{A(u)\le\alpha} S(N(R(x,u),\propto)) \quad \wedge \quad \inf_{A(u)>\alpha} S(N(R(x,u),A(u)); \forall\, x \epsilon\, U \text{— eq2.1}$$

Here we specify Lukasiewicz's T-norm T_L, then its dual conorm with respect to Ns is the bounded sum

$$S_L(x,y) = \min(1, x+y);$$

The continuous T-conorms include the standard max operator (the smallest T-conorm)

$$S_M(x,y) = \max(1, x+y).$$

- Inf represents infimum
- Infimum = minimum
- Sup represents supremum
- Supremum = maximum
- N is the negator that represents 1-x
- $A(u) \le \alpha$ when this condition satisfy take only those condition attribute value from decision table whose values are less than alpha.
- $A(u) > \alpha$ take only those attribute from decision table whose values are greater than alpha.
- α is a threshold value α ε [0,1]. Threshold is calculated $\alpha = (max + \min)/2$.
- x is objects of dataset.
- u is the attribute value of dataset.

3 Fuzzy Information System

In section-2 the basic concepts of fuzzy rough set approach has been discussed. This section describes about the Fuzzy Information System and overall process of proposed system.

Let U be a nonempty set with finite objects. With every object, we associate a set of condition attributes R and set of decision attributes D. Condition attributes having real number values and decision attribute having decision values. The pair$(U, R \cup D)$ is called a decision table, represented by DS. If some attribute in DS are fuzzy attributes the decision table is called fuzzy decision table, represented by FD =$(U, R \cup D)$. Here, fuzzy attributes mean the attribute with real number values since these values can be transferred to fuzzy values using membership degree. An example demonstrates the fuzzy information system. A decision table (DS) is show in Table 1 is a two-dimensional data table with 7 objects U={X1 ,X2 , X3, X4, X5, X6, X7}. The set of condition attributes R = {a, b, c, d, e, f, g, h, j} and one symbolic decision attribute D. Decision attributes D having two decision classes: 0 and 1. The objects {X1, X4, X5, X7) belongs to decision class 1 and {X2, X3, X6} belongs to decision class 0.

Table 1. Decision Table

Obj	a	b	c	d	e	f	g	h	j	D
X1	.523	.057	.047	.104	.090	.287	.01	.045	.69	1
X2	.603	.164	.916	.448	.986	.443	.358	.37	.247	0
X3	.918	.449	.283	.520	.642	.617	.994	.015	.407	0
X4	.221	.061	.041	.115	.083	.176	.054	.03	.053	1
X5	.037	.035	.074	.099	.009	.286	.095	.022	.069	1
X6	.429	.216	.334	.174	.634	.298	.522	.024	.195	0
X7	.215	.088	.019	.007	.055	.146	.07	.012	.178	1

3.1 System Overview

The overall process of the proposed system is shown in Figure-1and discussed in following steps.

Step 1: Take input dataset in real number values.
Step 2: Normalize the dataset using min-max normalization method.
Step 3: Form decision table from normalize dataset.
Step 4: Decision Table is randomly split it in to two folds (Training set 50% and Test set 50%)
Step 5: Calculate lower approximation operator
Step 6: Calculate the constant degree.
Step 6: Build the discernibility vector algorithm.

Step 7: Find the attribute value core of every object.
Step 8: Build the attribute value reduction algorithm.
Step 9: Attribute value reduction algorithm is used to generate classification rules.
Step 10: Classification rules are applied on testing dataset.
Step 11: Classification accuracy is measured.
Step 12: Construct classifier by transforming the generated rules in to program.
Step 13: Use the generated classifier to find new Objects with unknown class

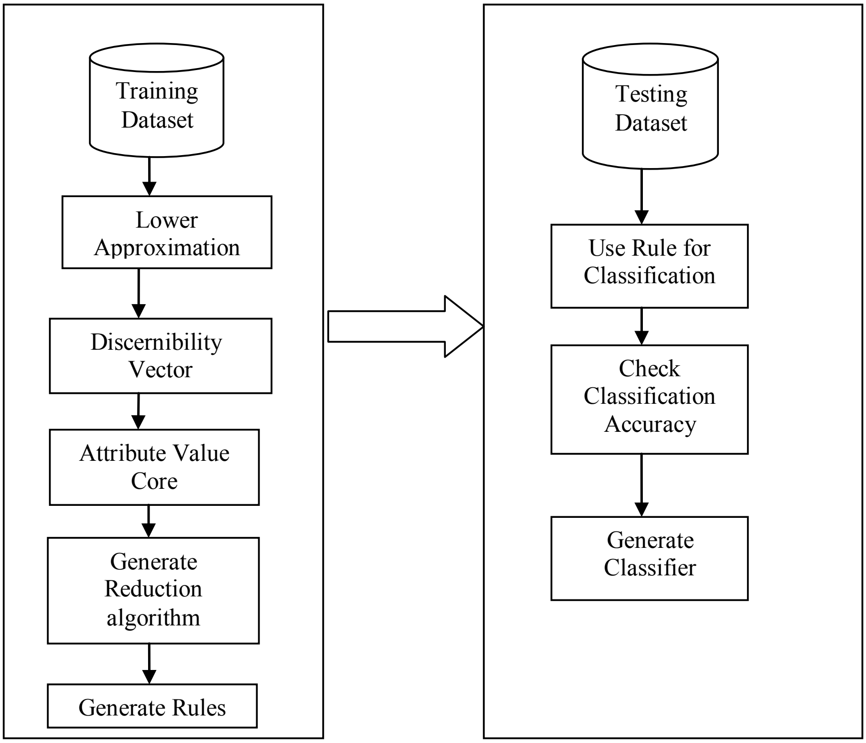


Fig. 1. System Overview Of Fuzzy Rough Set Approach Based Classifier

Table 2. The Lower Approximation Value and Constant Degree

Obj	X1	X2	X3	X4	X5	X6	X7
Lower Approximation for D=1	.523	0	0	.779	.713	0	0
Lower Approximation for D=0	0	.551	.521	0	0	.785	.594
Constant Degree(CD)	.523	.551	.521	.779	.713	.785	.594

Firstly dataset is normalized using min-max normalization method and decision table is formed. We calculate the lower approximation operator using the formulae as described in eq-2.1

4 Discernibility Vector

In the previous section we have calculated lower approximation value and constant Degree. Now we calculate discernibility vector to delete redundant attributes and get attribute value core of every object. The discernibility vector is generated shown in Table-2.

Compute discernibility vector Dv_{ij} using the following steps:

1. Decision table is read as input
2. We find the objects which are belonging to the same decision class and when objects are equal set ∅ such as {x1,x4,x5,x7} belongs to decision class 1 and objects {x2,x3,x6} belongs to decision class 0.
3. The objects belongs to different decision class such that $D(x1) \neq D(x2)$. We find the difference between the two objects, compare it with constant degree. We only take those attribute from fuzzy decision table-1 whose difference between the objects is greater than equal to the constant degree, if the difference between the object is negative it will be considered as positive only. Such as

```
Compute R
Compute $CD(x_i)$ where $CD(x_i)=m$
 if($D(x_i)==D(x_j)$ OR ( $x_i == x_j$ )
 $Dv_{ji}$ ={ }
if ($D(x_i) \neq D(x_j)$)
      $Dv_{ji}$ = abs{(a:N(a($x_i, x_j$) $\geq$ m)}
```

Table 3. Discernibility Vector

Obj	X1	X2	X3	X4	X5	X6	X7
X1	{}	{c, e}	{e, g}	{}	{}	{e}	{}
X2	{}	{}	{}	{c, e}	{c, e}	{}	{c,e}
X3	{c,e}	{}	{}	{g}	{a,g}	{}	{g}
X4	{e,g}	{c,e,j}	{a,e, g,j}	{}	{}	{e}	{}
X5	{}	{a,c,e,j}	{a,e,g,j}	{}	{}	{e}	{}
X6	{e}	{}	{}	{}	{}	{}	{}
X7	{}	{c,e,j}	{a,e,g,j}	{}	{}	{e}	{}

4.1 Attribute Value Core of Every Object

The core may be thought as set of necessary attributes. It is possible that core may be empty, which means that there is no indispensable attribute. The attribute value core is the collection of the most important attributes value to distinguish one object from others. The attributes value core is the collection of single attribute in discernibility vector. The attribute Value Core is generated shown in Table-4.

Table 4. Attribute Value Core Of Every Object

Obj	X1	X2	X3	X4	X5	X6	X7
Core	{e,.090}	{ }	{ }	{g,.054}	{ }	{e,.634}	{g,.07 }

Now the Attribute value reduction algorithm is described. It keeps only those attribute which are necessary. The attribute value reduction is generated shown in Table-5.

Attribute Value Reduction Algorithm

```
1.  Find the first attribute from the discernibility
    vector of each row and column.
2.  Form the set of same row and column.
3.  If  size of the both set is equal
    C1=$\{set1(attribute) \wedge set2(attribute)\} \vee$
    $\wedge \{(set1(attribute) \vee set2(attribute)) - set1(attribute) \wedge$
    $set2(attribute)\}$.
4.  Take value of those attribute which comes in C1
    from decision table-1.
5.  In C1 one more attribute is added from table-4
    attribute value core  $C1 \vee (attribute\ value\ core)$.
6.  If size of both set is not equal then find only
    common attribute  among those set.
    $C2 = \{(set1(attribute) \wedge set2(attribute)\}$
7.  In this set add one more attribute from table-4
    attribute value core.
    $c2 \vee attribute\ value\ core$.
```

Table 5. Attribute Value Reduction

Obj	a	b	c	d	e	f	g	h	j	D
X1	*	*	.047	*	.09	*	*	*	*	1
X2	*	*	.916	*	*	*	*	*	*	0
X3	.918	*	*	*	*	*	*	*	*	0
X4	*	*	.041	*	*	*	.054	*	*	1
X5	.037	*	*	*	.009	*	*	*	*	1
X6	*	*	*	*	.634	*	*	*	*	0
X7	*	*	.01	*	*	*	.07	*	*	1

5 Rule Induction from Attribute Value Reduction Table

In Previous Section attribute value reduction algorithm was generated. In this section this algorithm is used to generate rules. These rules are as follows:

```
1) Compare lower approximation value with maximum value
   in Decision table-1
       1.1 If lower approximation value is greater than
      equal to maximum Value in decision table then D=0.
2) If lower approximation value is less than maximum
   value in decision table    then D=1.
```

5.1 Experimental Results

In this section three dataset Ionosphere, Bupa, Monk from UCI Machine Learning Repository [7] are used for classification purpose, implemented using JAVA. Classification rule set is generated using half of the sample dataset objects, remaining dataset objects are used for the purpose of testing to check the accuracy. Classification rule set is applied on the testing dataset and decision class is generated, this is compared with the actual decision class of the objects. Classifier is correctly classifying the objects if there is match with the actual decision class. The basic information of these datasets is summarized in Table 6.

Table 6. Basic Information About Dataset

Datasett	Abberviation	Objects	Data Type (Condition Attributes)	Attributes	Decision Class
1.	Ionosphere	351	Real number	34	2
2.	Bupa	345	Real Number	7	2
3.	Monk	100	Real Number	7	2

5.2 Comparison of Accuracy with Fuzzy Rough Set Approach Based Classifier

In Fuzzy rough set approach based classifier we have taken benchmark of Ionosphere, Bupa, Monk dataset and we have found that no of rules are reduced and accuracy is improved as compare to [1] ionosphere approximate 12.5%. Monk, Bupa dataset are also taken under consideration, we have proved that the accuracy of these benchmark using fuzzy rough set approach based classifier is 66%, 52.5% respectively.

Table 7. Result on dataset

Dataset	Number Of Rules GFRSC	Accuracy
Ionosphere	6.619	78%
	7.2318	78%
	6.7619	77%
	6.9048	77%
	8.0952	77%

Table 8. Improvement in Accuracy

Serial Number	Dataset	Number of rules FRSC	Accuracy
1.	Ionosphere	2	90.5%
2.	Bupa	2	52.5%
3.	Monk	2	66%

6 Conclusion

In this paper, we develop a fuzzy rough set based classifier. The key idea of developing fuzzy rough set based classifier is to reduce complexity of dataset, training time of classifier, improved performance in terms of accuracy. First we calculate lower approximation, to reduce redundant attribute value, for calculating lower approximation we compare each value of decision table with threshold value. Threshold is kept constant so that each value comes in precise range then discernibility vector designed and attribute value reduction algorithm is developed, to extract new classification rules. Experimental results shows that number of rules are reduced and accuracy is improved on benchmark of dataset.

References

[1] Zhao, S., Tsang, E.C.C., Chen, D., Wang, X.: Building a Rule-Based Classifier—A Fuzzy-Rough Set Approach. IEEE Transaction on Knowledge and Data Engineering 22, 624–634 (2010)

[2] Yeung, D.S., Chen, D., Tsang, E.C.C., Lee, J.W.T., Wang, X.: On the Generalization of Fuzzy Rough Sets. IEEE Transcation on Fuzzy Systems 13(3) (June 2005)

[3] Yasdi, R.: Learning Classification Rules from Database in the Context of Knowledge Acquisition and Representation. IEEE Trans. Knowledge and Data Eng. 3(3), 293–306 (1991)

[4] Wang, X.Z., Tsang, E.C.C., Zhao, S.Y., Chen, D.G., Yeung, D.S.: Learning Fuzzy Rules from Fuzzy Samples Based on Rough Set Technique. Information Sciences 177(20), 4493–4514 (2007)

[5] Wu, W.Z., Mi, J.S., Zhang, W.X.: Generalized Fuzzy Rough Sets. Information Sciences 151, 263–282 (2003)

[6] Wu, W.Z., Zhang, W.X.: Constructive and Axiomatic Approaches of Fuzzy Approximation Operators. Information Sciences 159, 233–254 (2004)

[7] Blake, C., Keogh, E., Merz, C.J.: UCI Repository of Machine learning Databases, Univ. California Irvine, `http://www.ics.uci.edu`

Proposing a CNN Based Architecture of Mid-level Vision for Feeding the WHERE and WHAT Pathways in the Brain

Apurba Das[1], Anirban Roy[2], and Kuntal Ghosh[3]

[1] Centre for Development of Advanced Computing (CDAC), Salt Lake, Kolkata, India
[2] Techno India, Salt Lake, Kolkata, India
[3] Machine Intelligence Unit and Center for Soft Computing Research,
Indian Statistical Institute
apurba.das@cdac.in,
anirbanroy.ti@gmail.com, kuntal@isical.ac.in

Abstract. In the central visual pathway originating from the eye, a bridging is required between two hierarchical tasks, that of pixel based information recording by visual pathway at low level on one hand and that of object recognition at high level on the other. Such a bridge which may be designated as a mid-level block-grained integration has here been modeled by a multi-layer flexible cellular neural network (F-CNN). The proposed CNN architecture is validated by different intermediate level tasks involving rigid and deformable pattern recognition. Execution of such tasks by the proposed architecture, it has been shown, is capable of generating valid and significant inputs for the WHERE (dorsal) and WHAT (ventral) pathways in the brain. The model includes the proposal of a feedback (also by CNN architecture) to the lower mid-level from the higher mid-level dorsal and ventral pathways for flexible cell (physiological receptive field) size adjustment in the primary visual cortex towards successful 'where' and 'what' identifications for high-level vision.

Keywords: Cellular Neural Network (CNN), dorsal and ventral pathways, visual cortical column, scale-space, attention model.

1 Introduction

Object recognition is a high-level vision task of the human brain which comprises of identifying the object of interest as well as locating it in space, the two tasks that are dealt by the parallel ventral and dorsal pathways respectively. The neuro-visual system initially generates an inner representation of the external stimuli, enabling the brain to construct internal data structures for recording the spatial distribution of stimuli attributes, which eventually become the resources that are processed possibly by a range of perceptual algorithms. An appropriate modeling of visual system of human brain can significantly enhance the efficiency of computer based image processing. Since the information contained in a single pixel is very limited, so the semantic content of any image must be embodied by stage-by-stage integration of a

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 559–568, 2011.
© Springer-Verlag Berlin Heidelberg 2011

number of pixels. David Marr [1] pointed out in his path-breaking work that at least three levels are involved in this type of processing: a physical representation, with a grain size of a pixel, a middle representation with a grain size of a 'meaningful' block that involves some basic knowledge about color, intensity and texture and finally, a semantic representation of an object, involving knowledge about its structure, 3D pose and background. It is quite understandable that the mid-level representation (2.5D representation [1]) plays a very significant role in inheriting the product of the initial stage of processing, and ushering in the final stage. Hence it is likely that in the middle level processing, a prior knowledge about two-class vision (object and background) is required from the high level of processing to make a meaningful block-grained integration of pixels. A classification (perceptual grouping) may therefore be performed in the primary layer of the mid-level vision system in terms of figure-ground segregation. In traditional digital image processing, these operations may do no more than to differentiate and label pixels by locating the demarcating edges. Retinal ganglion cell based Classical receptive field (CRF) kernels (like the Laplacian of Gaussian (LOG)) have long been used to this end. For the next stage, researchers [2] have distinguished two major pathways for the cortical processing of visual information: the ventral visual pathway, for identifying objects (WHAT), and the dorsal visual pathway, for determining their positions in space (WHERE).

In the present work, we have proposed a cellular neural network (CNN) based architecture of the primary visual cortex to realize the mid-level vision. In section 2 and 3 the figure-ground segregation and attention models have been realized by CNN, respectively. The proposed architecture produced results which promise some useful inputs for the dorsal and ventral pathways of high level vision system in terms of position determination and identification of early patterns, as has been discussed in section 4. The Conclusion section identifies future directions of research.

2 CNN Based Architecture for Lower Mid-level Vision

In biological visual systems, each neuron only responds to stimuli in a certain part of the visual field, and this area is referred to as the classical receptive field (CRF) of a neuron. The CRF is sensitive to brightness and contrast, and has an antagonistic center-surround structure that is typically modeled by a Difference of Gaussian (DOG) or a Laplacian of Gaussian (LOG) ([3], [1]) (Fig. 1).

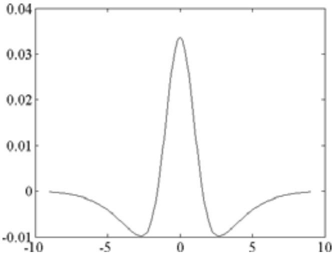


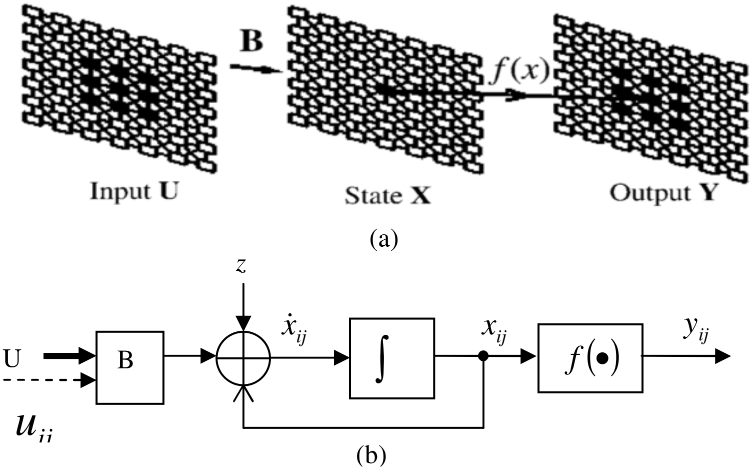
Fig. 1. One-dimensional profile of well-known classical receptive field represented by DOG

A standard cellular neural network (CNN) architecture [4], on the other hand, consists of an $M \times N$ rectangular array of cells $(C(i, j))$ with Cartesian coordinates (i, j). The representation is very suitable for image representation and processing. In addition to image processing, we can exploit its unique brain-like architecture to implement the receptive field based brain-like information processing tasks, where conventional ANN have been found wanting. Here we propose a zero feedback CNN (ZFCNN)[4] to first realize the lower mid-level operation of the visual cortex of human brain. The task may be looked upon as that for figure-ground segregation, where the object of interest needs to be segregated out from the background.

Each cell of the ZFCNN [4] is characterized by the following differential equation:

$$\dot{x}_{ij} = -x_{ij} + B * U_{ij} + z \tag{1}$$

$$y_{ij} = f(x_{ij}) \tag{2}$$



(a)

(b)

Fig. 2. Zero Feedback Cellular Neural Network $CNN \in C(0, B, z)$. (a) Signal flow structure of a ZFCNN with 3×3 neighborhood. (b) System structure of a cell.

The pathway from U to X symbolizes weighted contribution of the input cells to the center cell, i.e., convolution. Contribution from the input synaptic operator $B(i, j; k, l)$ is defined by

$$\begin{bmatrix} b_{-1,-1} & b_{-1,0} & b_{-1,1} \\ b_{0,-1} & b_{0,0} & b_{0,1} \\ b_{1,-1} & b_{1,0} & b_{1,1} \end{bmatrix} * \begin{bmatrix} u_{i-1,j-1} & u_{i-1,j} & u_{i-1,j+1} \\ u_{i,j-1} & u_{i,j} & u_{i,j+1} \\ u_{i+1,j-1} & u_{i+1,j} & u_{i+1,j+1} \end{bmatrix} = B * U_{ij} \tag{3}$$

In the Eq. 1, U, B and X represents input image, the feed forward input cloning mask and state of operation respectively. The integrator works as an accumulator which forms the entire convolved image accumulating the state operations (shifting of convolution mask

B). Zero feedback signifies zero coupling from output to input stage. In our proposal, the size of the mask (B) is flexible. The size would be adjusted from the feedback of the next CNN layer which we designate as the higher mid-level where we would design the architecture depending on the attention model proposed by Koch and Itti [5], while the present section concentrates on designing the CNN equivalent model of the lower mid-level vision task, which essentially suggests the segregation of the figure of interest from the background. It is convenient to decompose the template B as follows.

$$B = B_0 + \overline{B} \tag{4}$$

where,

$B_0 =$

0	0	0	0	0
0	0	0	0	0
0	0	4n-4	0	0
0	0	0	0	0
0	0	0	0	0

$\overline{B} =$

-1	-1	-1	-1	-1
-1	0	0	0	-1
-1	0	0	0	-1
-1	0	0	0	-1
-1	-1	-1	-1	-1

where n is the size of the mask. In the above representation the mask size is 5×5. So, $b_{0,0}=4\times5-4=16$. It is to be noted that, in the design of the mask summation of all the elements is essentially zero so that, the operation does not contribute to the natural image as noise by modifying the average intensity.
That is,

$$\sum_{\forall i}\sum_{\forall j} b_{i,j} = 0 \tag{5}$$

also,

$$b_{0,0} = -\sum_{\forall i}\sum_{\forall j} \overline{b}_{i,j} \tag{6}$$

Here, the bias z is fixed at zero, and the activation function of the CNN $f(\bullet)$ is a *signum* function. For $x_{ij} \le 0$, the output $y_{ij} = -1$, while for positive x_{ij}, the output $y_{ij} = 1$. So the output is essentially a binary image consisting of the figure group and the background group respectively.

3 CNN Based Architecture for Higher Mid-level Vision

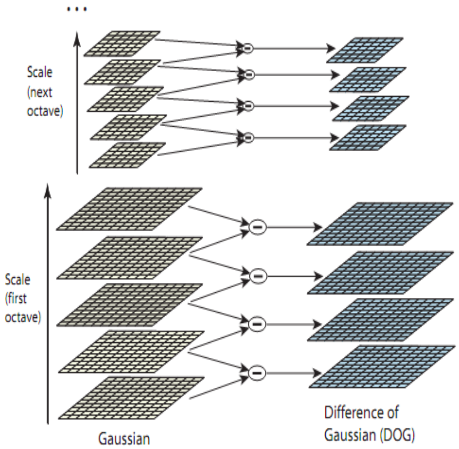
To realize the next level of vision we propose our next layer of CNN module to make more meaningful the afore-mentioned grained-block integration of the image. In the scale space concept of attention model, proposed by Koch and Itti [5], the input image to the central visual pathway passes through three parallel pyramids one each for color, intensity and orientation information extraction. In our proposed CNN based architecture we have considered fixed intensity, orientation and gray color space.

3.1 Scale-Space Pyramidal Representation

In our proposed architecture, we have formed the image pyramids depending upon the scale-space theory. In each scale, the images are blurred by different Gaussian masks with linearly varied σ. To the next octave of the pyramid, under-sampling and blurring is done on each image for all scales. In each scale we have considered 5 images. Each pair of the images is then passed through two different activation functions corresponding excitatory and inhibitory neurons, respectively. Thus, four Difference of Gaussian (DoG) images are obtained in each scale [6, 7] as shown in Fig. 3 and 4.

3.2 Blob Detection from DoG in Primary Visual Cortex (V1)

In the next layer of CNN, the scale-space stable points are determined which showed maximum pixel intensity with respect to their eight neighbors in each scale of DoG decomposition. These scale invariant key-points are essentially centers of the blobs detected from scale-space decomposition of a particular image.



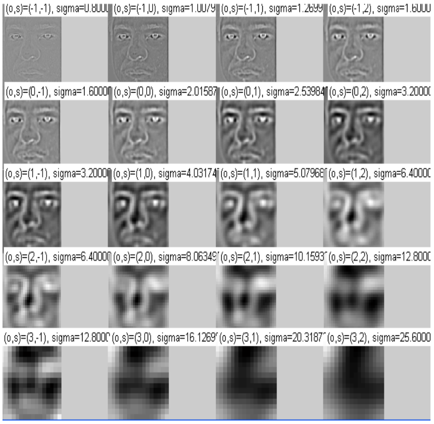
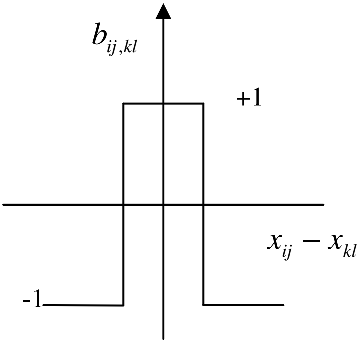

(o,s)=(-1,-1), sigma=0.800
(o,s)=(-1,0), sigma=1.0079
(o,s)=(-1,1), sigma=1.2699
(o,s)=(-1,2), sigma=1.6000
(o,s)=(0,-1), sigma=1.6000
(o,s)=(0,0), sigma=2.01587
(o,s)=(0,1), sigma=2.53984
(o,s)=(0,2), sigma=3.2000
(o,s)=(1,-1), sigma=3.2000
(o,s)=(1,0), sigma=4.03174
(o,s)=(1,1), sigma=5.0796
(o,s)=(1,2), sigma=6.4000
(o,s)=(2,-1), sigma=6.4000
(o,s)=(2,0), sigma=8.0634
(o,s)=(2,1), sigma=10.1593
(o,s)=(2,2), sigma=12.800
(o,s)=(3,-1), sigma=12.800
(o,s)=(3,0), sigma=16.1269
(o,s)=(3,1), sigma=20.3187
(o,s)=(3,2), sigma=25.6000

Fig. 3. Subtraction between two adjacent smooth images to compute DoG

Fig. 4. DoG on the scale space



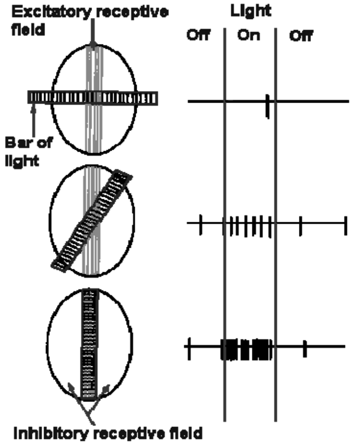


Fig. 5. Transfer mask of proposed CNN layer for extrema detection

Fig. 6. Importance of orientation in visual cortex

As recorded by Livingstone and Hubel [8, 9], the blobs detected in the V1 elongated cells of the primary visual cortex projects to the V2 cells. The CNN mask for extrema detection is shown in the Fig. 5, for two states of network x_{ij}, x_{kl} referred as neighboring pixels of the DoG image.

3.3 Realization of Orientation Column in Visual Cortex by CNN

The receptive fields of cortical simple cells closest to the ganglion field consist of extended narrow central portions flanked by two antagonistic areas. The central area is excitatory and the surround is inhibitory (Fig. 6). For optimal activation, it needs a bar of light of a certain width, entirely filling the central area of the receptive field of a simple cell in the striate cortex (V1), and oriented at a certain angle. This is shown in the Fig. 6, where for a deviation from the preferred orientation the response of the receptive field deteriorates [8, 9].

The well-known function of visual cortex namely orientation selectivity [10] can be realized by CNN. For instance, we demonstrate how the uncoupled cloning template shown below detects bars with a –45° slope. The geometry of the positive terms in the B template determines the enhancement of the derivative of the state variable and the positive feedback brings the values to black and white.

$$\mathbf{A} = \begin{array}{|c|c|c|}\hline 0 & 0 & 0 \\ \hline 0 & 2 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} \qquad \mathbf{B} = \begin{array}{|c|c|c|}\hline 0.25 & 0 & 0 \\ \hline 0 & 0 & 0 \\ \hline 0 & 0 & 0.25 \\ \hline \end{array} \qquad z = -1$$

The CNN here can be modeled as

$$\dot{x}_{ij} = -x_{ij} + A * Y_{ij} + B * U_{ij} + z \tag{7}$$

$$y_{ij} = f\left(x_{ij}\right) \tag{8}$$

Here the mask A is designed taking higher weight in the middle, so that orientation assignment is forced to be done around the detected extrema points only. In this way, the local image gradients are measured and represented at the selected scale in the region around each keypoint. The key-points with magnitudes and orientations are shown in Fig. 7 for generated binary image.

3.4 Feature Descriptor Design for Position and Identity Detection

The descriptor is formed from a vector containing the values of all the orientation histogram entries, corresponding to the lengths of the arrows as shown in the Fig. 9. From the computed gradient of each key-point the 4×4 neighborhood, we have considered 8 orientation bins. Hence feature descriptor of length 4×4×8=128 is formed for each key-point [11, 12]. As seen in Fig. 9, the CNN itself can identify and detect the proper position of the figure of interest in presence of scale and affine rotation (the book) in terms of matching the descriptors of the points. If the accumulated matching score between each pair of feature points is higher than the predefined threshold ($z=0.5$), we declare them a matched pair of points. Thus visual cues for both WHERE and WHAT pathways have been extracted by proposed CNN. The images of Fig. 9 are taken from dataset created in Koch lab [13]. The relative spatial occupancy of the object (book) is larger in the left figure. The relative position and neighborhoods are also quite different. Yet we have found significant match (65 matched pairs) when the two binary images obtained from the primary layer of CNN for figure-ground segregation using mask sizes 43 and 53 for the left and right image respectively. For mask size 43 and 63 (left image), number of matches obtained is reduced from 65 to 58 and 60 respectively. Therefore, possible inputs to the WHERE (location) and WHAT (identity) pathways in the central visual organization can be maximized by fixing this mask size. Detailed discussion follows in the next section.

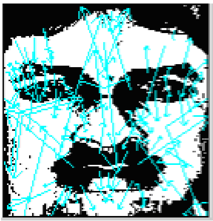


Fig. 7. Keypoint magnitude and orientation assignment on lower mid level of visual cortex

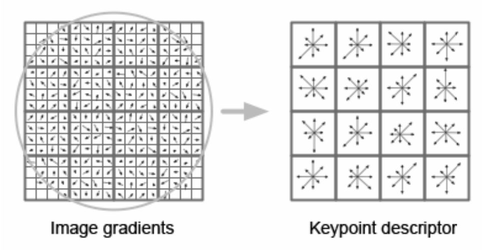


Fig. 8. Key-point descriptor generation from gradient magnitude and orientation

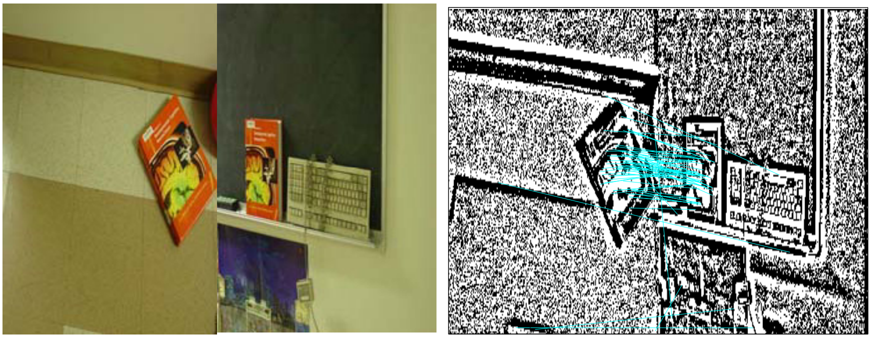

(a) (b)

Fig. 9. Rigid object recognition by flexible CNN model of visual cortex

4 Maximizing Stimulations to WHERE and WHAT Pathways

Following the path-breaking studies by Ungerleider and Mishkin [2], scientists distinguished two major pathways (roughly parallel) for the cortical processing of visual information: the ventral pathway, for identifying objects (WHAT), and the dorsal pathway, for determining their position in space (WHERE). The dorsal pathway could thus be described as an ‘action pathway’, because it lets us interact with the environment effectively. The ventral pathway, on the other hand, seems to be involved in forming conscious representations of the identity of objects.

The encouraging results in the previous section towards constructing possible feeds for the dorsal and ventral pathways in terms of the number of matched pairs now motivate us to make provisions for mask size adjustment from the proposed higher mid- to lower mid-level vision. We have tested the proposed model of mid-level vision on different types of figure-ground segregated images including rigid (Fig. 9) and deformable (Fig. 10) objects. We found the proposed CNN system robust to unconstrained scale and orientation varied environment.

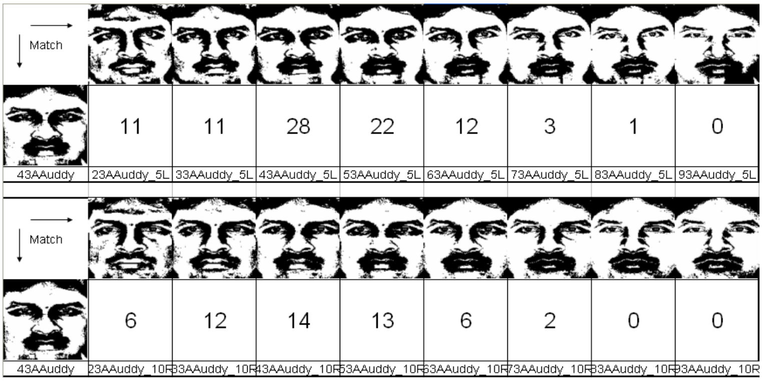
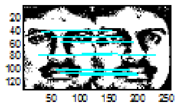
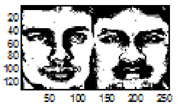
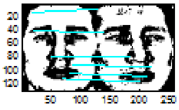
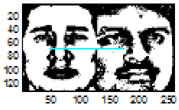


Fig. 10. Number of matched points between the frontal and posed faces with varying mask

By taking one frontal and the rest pose images of 5 degree left and 10 degree right of the same person with size 128×128 (Fig. 10), it is observed that when the mask size exactly matches, the recognition score in terms of matched key-points is maximized. It can be considered as a preprocessor for processing by dorsal and ventral pathways. Thus the varied mask size CNN is named as flexible CNN (F-CNN) and the flexibility is obtained by a feedback from our proposed higher mid-level to lower mid-level of visual cortex.

Table 1. Face Matching by Multilayer F-CNN

Matched pair	Unmatched pair
 10 matches (F to 10L)	 0 matches (10 L to F)
 6 matches (F to 10L)	 1 match (F to 10L)

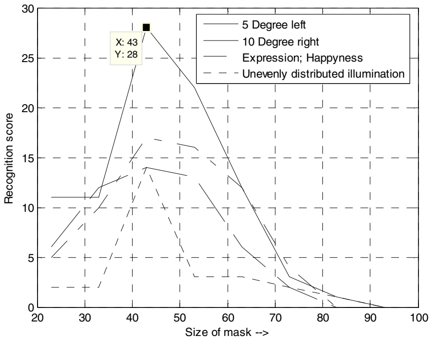


Fig. 11. Maximum matched points obtained for identical relative mask size

On the other hand, the proposed architecture also ensures that even for similar mask size, the number of matched pairs between face images of different subjects is poor. It ensures reduction of false acceptance and false rejection thus strengthening the decision making capabilities of the two parallel visual pathways. The mask size can therefore be automatically varied by the feedback driven by the recognition score (Yij) with feedback transfer function A, here defined as flexibility parameter, which changes the input mask size B of the first layer of the CNN.

The proposed F-CNN is used for maximizing the accuracy of the two pathways by increasing the number of matched points between peer face images (same subject with PIE variation) and rejecting wrong pair of face images (different subjects) successfully as tabulated in Table 1. Here all the face images are normalized to 128×128 pixels, so when the flexile mask size of the input layer of the CNN converges to the mask size of the peer match of PIE varied face image, then maximum number of identical key points are generated indicating enhancement of the feed forward paths in the central visual pathway as shown in Fig. 11.

5 Conclusion

The proposed CNN based architecture model is inspired from the functionality of mid-level vision. Early signatures of patterns can be interpreted by the proposed network. The experimental results are quite in agreement with the fact that the middle level visual system feeds to the WHERE and WHAT visual pathways significantly. In future, the relative size of the pattern of interest present in an image could be taken care of by the feedback to make the proposed CNN further flexible. Also, attempts may be made to model the depth information so that 3D recognition is made possible by the CNN based architecture. The dichotomy between the conscious (WHAT) and unconscious (WHERE) modules in visual processing can also be addressed in future using the proposed network.

References

1. Marr, D.: Vision: A computational investigation into the human representation & processing of visual information. MIT Press (2010)
2. Ungerleider, L.G., Mishkin, M.: Two Cortical Visual Systems. In: Ingle, D.J., Goodale, M.A., Mansfield, R.J.W. (eds.) Analysis of Visual Behavior, pp. 549–586. The MIT Press, Cambridge (1982)
3. Rodieck, R.W., Stone, J.: Analysis of receptive fields of cat retinal ganglion cells. Journal of Neurophysiology 28, 833–849 (1965)
4. Chua, L.O., Roska, T.: Cellular Neural Networks and Visual Computing. Cambridge University Press (2002)
5. Itti, L., Koch, C., Niebur, E.: A model of saliency based visual attention for rapid scene analysis. IEEE Trans. on PAMI 20, 1254–1259 (1998)
6. Koenderink, J.J.: The structure of images. Biological Cybernetics 50, 363–396 (1984)
7. Lindeberg, T.: Scale-space theory: A basic tool for analyzing structures at different scales. Journal of Applied Statistics 21(2), 224–270 (1994)

8. Livingstone, M.S., Hubel, D.H.: Anatomy and physiology of a colour system in the primate visual cortex. J. Neurosci. 4, 309–356 (1984)
9. Livingstone, M.S., Hubel, D.H.: Segregation of form, colour, movement, and depth: anatomy, physiology, and perception. Science 240, 740–749 (1988)
10. Kandel, E.R., Schwartz, J.H., Jessel, T.M.: Principles of Neural Science, 3rd edn. Elsevier, New York (1991)
11. Lowe, D.G.: Object recognition from local scale-invariant features. Proceedings of the International Conference on Computer Vision 2, 1150–1157 (1999)
12. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Key points. International Journal of Computer Vision 60(2), 91–110 (2004)
13. `http://www.klab.caltech.edu/codedata/codedata.shtml`

Multithreaded Memetic Algorithm for VLSI Placement Problem

Potti Subbaraj[1] and Pothiraj Sivakumar[2]

[1] Sri Nandhanam College of Engineering & Technology, Tirupattur, Vellore District,
TamilNadu - 635601, India
`subbaraj_potti@yahoo.com`
[2] Electronics and Communication Engineering Department,
Arulmigu Kalasalingam College of Engineering, Krishnankoil,
Tamil Nadu - 626126, India
`siva@klu.ac.in`

Abstract. Due to rapid advances in VLSI design technology during the last decade, the complexity and size of circuits have been rapidly increasing, placing a demand on industry for faster and more efficient CAD tools. Physical design is a process of converting the physical description into geometric description. Physical design process is subdivided into four problems: Partitioning, Floor planning, Placement and Routing. Placement phase determines the positions of the cells. Placement constrains are wire-length, area of the die, power minimization and delay. For the area and wire length optimization a modern placer need to handle the large–scale design with millions of object. This thesis work aims to develop an efficient and low time complexity algorithms for placement. This can be achieved by the use of a problem specific genotype encoding, and hybrid, knowledge based techniques, which support the algorithm during the creation of the initial individuals and the optimization process. In this paper a novel memetic algorithm, which is used to solve standard cell placement problem is presented. These techniques are applied to the multithread of the VLSI cell placement problem where the objectives are to reduce power dissipation and wire length while improving performance (delay).

Keywords: VLSI design, physical design, placement, standard cell, multithread.

1 Introduction

In the past thirty years, VLSI-CAD (Computer-Aided Design of Very Large-Scale Integrated circuits) has been an enabling force behind the exponential growth of the performance and capacity of integrated circuits [6]. The most common way of breaking up the layout problem into sub-problems is first to do logic partitioning where a large circuit is divided into a collection of smaller modules according to some criteria, then to perform component placement, and finally determine the approximate course of the wires in a global routing phase after which a detailed-routing phase determines the exact course of the wires in the layout area. The main objectives of the placement design process are to minimize the total chip area and the total estimated wire length for all the nets. Optimization of the chip area usage can fit more functionality into a given chip area. Optimization of the total estimated wire length

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 569–576, 2011.
© Springer-Verlag Berlin Heidelberg 2011

can reduce the capacitive delays associated with longer nets and speed up the operation of the chip. Thus, the placement design process has a pronounced affect on the final chip performance. Since module placement is an NP-hard problem, therefore, it cannot be solved exactly in polynomial time [5, 7]. Trying to get an exact solution by evaluating every possible placement to determine the best one would take time proportional to the factorial of the number of modules. Consequently, it is impossible to use this method for circuits with any reasonable number of modules. To search through a large number of candidate placement configurations efficiently, a heuristic algorithm must be used [2]. The traditional approach in placement is to construct a *global placement* (initial placement) by using constructive placement heuristic algorithms. A *detailed placement* follows to improve the initial placement. A modification is usually accepted if a reduction in cost occurs, otherwise it is rejected.

Global placement produces a complete placement from a partial or non-existent placement. It takes a negligible amount of computation time compared to detailed (iterative improvement) placement and provides a good starting point for them [20]. Usually, global placement algorithms include *random placement, cluster growth, partitioning-based placement* [9], *numerical optimization*, and *branch and bound techniques* [18]. Since partitioning-based methods and numerical optimization methods do not directly attempt to minimize wire length, the solution obtained by these methods is sub-optimal in terms of wire length. Therefore a detailed placement algorithm must be performed to improve the solution. There are two classes of detailed placement methods: *Deterministic* and *Stochastic heuristics*. A deterministic heuristic interchanges randomly selected pairs of modules and only accepts the interchange if it results in a reduction in cost [10].

2 Problem Formulation

A mathematical formulation of an approximation to the standard cell placement problem is now presented. Usually, a circuit is represented by a hypergraph *G (V, E)*, where the vertex set V = $\{v_1, v_2, v_3, \ldots, v_n\}$ represent the nodes of the hypergraph (set of cells to be placed), and E = $\{e_1, e_2, e_3, \ldots, e_n\}$ represents the set of edges of the Hypergraph (set of nets connecting the cells). The two dimensional placement regions are represented as an array of legal placement locations. The Hypergraph is transformed into a graph (a Hypergraph with all hyperedge sizes equal to 2) via clique model for each net. Each edge e_j is an order pair of vertices with a non-negative weight W_{ij} assigned to it. The placement task seeks to assign all cells of the circuit to legal locations such that cells do not overlap. Each cell *i* is assigned a location on the XY-plane. The cost of an edge connecting two cells *i* and *j*, with locations (x_i, y_j) and (x_i, y_j) is computed as the product of the squared l_2 norm of the difference vector $(x_i - x_j)$ $(y_i - y_j)$ and the weight of the connecting edge w_{ij} The total cost, denoted $\Phi(x,y)$ can then be given as the sum of the cost over all edges; i.e:

$$\Phi(\mathrm{x,y}) = \sum_{1 \le i < j \le n} w_{ij} \left[(x_i - x_j)^2 + (y_i - y_j)^2 \right] \tag{1}$$

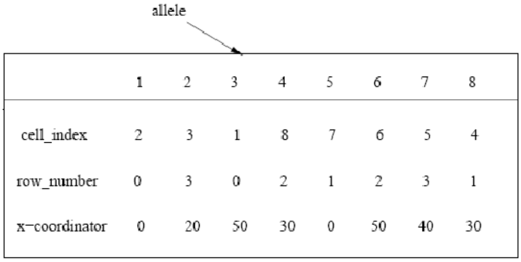
2.1 Wire Length Estimation

It is computationally expensive to determine the exact total wire length for all the nets at the placement stage and therefore, the total wire length is approximated during placement. To make a good estimate of the wire length, we should consider the way in which routing is actually done by routing tools. Almost all automatic routing tools use Manhattan geometry; that is, only horizontal and vertical lines are used to connect any two points. Further, two layers are used; only horizontal lines are allowed in one layer and only vertical lines in the other [20].

An efficient and commonly used method to estimate the wire length is the *semi-perimeter method* [19]. The wire length in this method is approximated by half the perimeter of the smallest bounding rectangle enclosing all the pins. For Manhattan wiring, this method gives the exact wire length for all two-terminal and three-terminal nets, provided that the routing does not overshoot the bounding rectangle. For nets with more pins and more zigzag connections, the semi-perimeter wire length will generally be less than the actual wire length. Moreover, this method provides the best estimate for the most efficient wiring scheme, the Steiner tree. The error will be larger for minimal spanning trees and still larger for chain connections. In practical circuits, however, two and three terminal nets are most common. Thus, the semi-perimeter wire length is considered to be a good estimate [20].

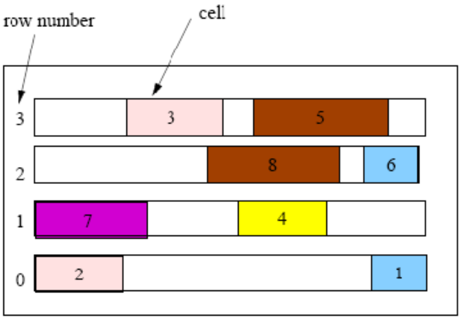
2.2 String Encoding

Genetic algorithms are powerful optimization techniques that have been successful in solving many hard combinatorial optimization problems [12]. The power of GA's comes from the fact that the technique is robust, and can deal successfully with a wide



	1	2	3	4	5	6	7	8
cell_index	2	3	1	8	7	6	5	4
row_number	0	3	0	2	1	2	3	1
x-coordinator	0	20	50	30	0	50	40	30

(a) String Encoding



(b) Placement

Fig. 1. String Encoding

range of problem areas, including those which are difficult for other methods to solve [3]. In this genetic-based placement algorithm, a string is represented by a set of alleles (the number alleles equal to the number of cells). Each allele indicates the index, the x-coordinates and row number of the cell. Fig.2a illustrates the string encoding of the cell placement given in Fig. 2b.

2.3 Scoring Function

Each individual is evaluated to determine its fitness through a scoring function. The scoring function F is the reciprocal of the total *HPWL* for all the nets.

$$F = \frac{1}{\frac{1}{2}\sum_{i=1}^{n} HPWL_i} \tag{2}$$

Where, *HPWL* is the sum of the half perimeter of the smallest bounding rectangle for each net. $HPWL_i$ is the estimate wire length of net i and n is the number of nets. In the implementation, cell overlaps are removed and row lengths are adjusted before evaluating the chromosome. One reason of doing so is due to the fact that removing the overlaps after every generation not only gives the algorithm a more accurate picture of the wire length but also gives the algorithm repeated chances to optimize the circuit after it has been perturbed by overlap removal [21]. Therefore the row length control and overlap penalty are not considered in the scoring function.

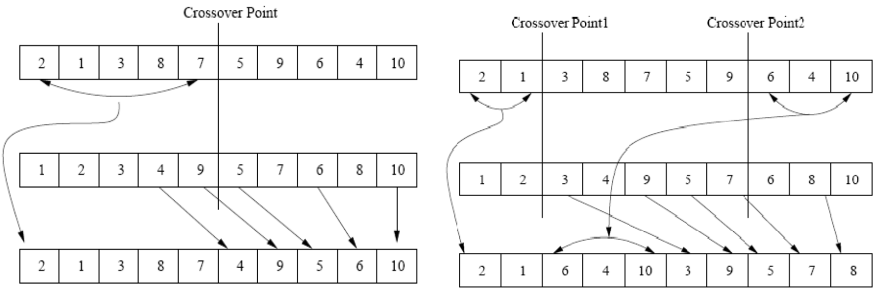
2.4 Initial Population Construction

For each configuration in the initial population, the x-coordinate and row numbers of cells are determined randomly. This kind of constructor can diversify the initial solutions but also tend to have a slower rate of convergence. Thus, some placement solutions produced by Cluster Seed method are injected into the initial population to increase the convergence rate.

2.5 Crossover and Mutation Operator

The Traditional crossover operator used in GA may produce infeasible solutions for the standard cell placement problem, therefore a crossover operator called Order crossover is considered.

Fig. 2a shows a one-point order crossover operator where each pair of parents generates two children with a probability equal to the crossover rate. In this method, a single cut point is chosen at random. The crossover operator first copies the array segment to the left point from one parent to one offspring. Then it fills the remaining part of the offspring by going through the other parent, from the beginning to the end and taking those elements that were left out, in order. The two-point order crossover operator is similar to one-point order crossover operator, except that it has to choose two crossover points randomly. An example of two-point order crossover operator is illustrated in Fig. 2b.



(a) One–Point Order Crossover (b) Two–Point Order Crossover

Fig. 2. One-Point and two-Point Order Crossover

Following crossover, each offspring is mutated with a probability equal to the mutation rate. The mutation operator mutates an individual by interchanging randomly selected pair of cells without changing the x-coordinate and row number. Replacement follows mutation, where the population for next generation is chosen from the combined set of parents and offspring's. This process is done by replacing the two worst individuals in the population with the offspring's only if the latter are better than the worst two individuals. This replacement method is also called "*elitism*" mechanism. One feature of this mechanism is that it ensures the maximum fitness of the population can never be reduced from one generation to the next.

2.6 Hybrid Simulated Annealing (SA)

In this work we have hybridized the genetic algorithm template with the SA method. The SA method is impregnated within GA, between the crossover and mutation operations, to improve all the solutions obtained after the crossover operation and before subjected to mutation operation. SA is very simple technique for State Space Search Problem. It can start from any state. And it is always move to a neighbor with the min cost (assume minimization problem). It can stop when all neighbors have a higher cost than the current state.

2.7 Multithreading

Multithreading (MT) is a technique that allows one program to do multiple tasks concurrently. The basic concept of multithreaded programming has existed in research and development labs for several decades. The threading models we describe are strictly software models that can be implemented on any general-purpose hardware. Much research is directed toward creating better hardware that would be uniquely suited for threaded programming. The memetic algorithm is modified slightly to make it distributed. A number of instances of the genetic algorithm are spawned and run independently an in parallel for number generations. After a set number of generations the separate instances stop and trade solutions with each other to introduce diversity into their populations and keep them from stagnating at local minima. They then repeat this process for a set number of epochs, which can be specified on the command line as well. After all epochs the best solution is chosen from all the instances of the genetic algorithm. To keep the separate instances from reaching the same local minimum only one crossover function is used per instance.

3 Experimental Results

In this work the proposed algorithm is coded in C++ to get a nearer to optimal solution for the VLSI placement problem and experiments were conducted in an Pentium Quad core 2 processor with 2.6 GHz. The outcome of results obtained for popular benchmark circuits ISPD02 was compared with that of the standard Genetic Algorithm methodology which is also coded in C++ and run on the same machine. The following parameters are fixed by conducting trials on various population sizes, crossover probabilities, mutation probabilities and the satisfactory convergent speed exploring scope of the proposed search techniques. Population size=20, Crossover probability (P_c) =0.6, Mutation probability (P_m) =0.01, Number of Threads=4. For all the bench mark circuit taken in this work, the proposed algorithm is able to outperform the standard Genetic algorithm both in terms of number of iterations required to reach a nearer to optimal solution.

Table 1. Comparison between Fast Place and MTMA

Circuit	Nets	Pins	HPWL(x10e6)		Run Time	
			Fast Place Capo	MTMA	Fast Place Capo	MTMA
ibm01	14111	50566	1.86	1.81	3m 59s	2m 50s
ibm02	19584	81199	4.06	4.00	7m 15s	5m 08s
ibm03	27401	93573	5.11	5.09	8m 23s	6m 19s
ibm04	31970	105859	6.39	6.35	10m 46s	8m 35s
ibm05	28446	126308	10.56	10.54	10m 44s	8m 38s
ibm06	34826	128182	5.50	5.45	12m 08s	8m 55s
ibm07	48117	175639	9.63	9.61	18m 32s	12m 15s
ibm08	50513	204890	10.26	10.15	19m 53s	14m 37s
ibm09	60902	222088	10.56	10.50	22m 50s	16m 15s
Ibm10	75196	297567	19.70	19.64	29m 04s	19m 58s
Ibm11	81454	280786	15.73	15.69	31m 11s	24m 01s
Ibm12	77240	317760	25.83	25.74	30m 41s	23m 11s
Ibm13	99666	357075	18.73	18.64	39m 27s	26m 24s
Ibm14	152772	548616	36.69	36.72	1h 12m	45m 03s
Ibm15	186608	715823	43.85	43.79	1h 30m	1h 11s
Ibm16	190048	778823	49.63	49.60	1h 31m	1h 12s
Ibm17	189581	860036	69.07	69.52	1h 43m	1h 58s
Ibm18	201920	819697	47.46	47.41	1h 44m	1h 59s

4 Conclusion

In this paper, we have presented multithreaded memetic algorithm for circuit placement and the performance was compared with Fast placement. Due to the ability of the Simulated Annealing method it finds the minimized wire length in fewer generations. We applied these algorithms to ISPD02 benchmark circuits from the results shows in Table 1, it is clear that MTMA outperforms than Fast Placement method. The experimental results show that multithreading has good accelerated performance on larger population sizes. A couple of other important factors can help to get the best performance out of our transformation to multithreading such as avoiding of excessive memory transfer, inherent parallelism and computation dependency. More sophisticated cluster approaches to parallelisation will have to be examined in the future.

References

1. Alpert, C.J., Kahng, A.B.: Netlist Partitioning: A Survey.: Integration. The VLSI Journal, 64–80 (1995)
2. Areibi, S.: Iterative Improvement Heuristics for the Standard Cell Placement: A Comparison. In: 5th World Multi Conference on Systemics, Cybernetics and Informatics, Orlando, Florida, pp. 89–94 (2001)
3. Areibi, S., Moussa, M., Abdullah, H.: A Comparison of Genetic/Memetic Algorithms and Other Heuristic Search Techniques. In: International Conference on Artificial Intelligence, Las Vegas, Nevada, pp. 660–666 (2001)
4. Areibi, S., Thompson, M., Vannelli, A.: A Clustering Utility Based Approach for ASIC Design. In: 14th Annual IEEE International ASIC/SOC Conference, Washington, DC, pp. 248–252 (2001)
5. Blanks, J.P.: Near Optimal Quadratic Based Placement for a Class of IC Layout Problems. IEEE Circuits and Devices 1(6), 31–37 (1985)
6. Chang, H., Cooks, L., Hunt, M.: Surviving the SOC Revolution. Kluwer Academic Publishers, London (1999)
7. Donath, W.E.: Complexity theory and design automation. In: 17th Design Automation Conference, pp. 412–419 (1980)
8. Etawil, H., Areibi, S., Vannelli, T.: Convex Programming based Attractor-Repeller Approach for Global Placement. In: IEEE International Conference on CAD, San Jose, California, pp. 20–24 (1999)
9. Garey, M.R., Johnson, D.S.: Computers and Intractability. Freeman, San Francisco (1979)
10. Goto, S., Kuh, E.: An approach to the two-dimensional placement problem in circuit layout. IEEE Trans., Circuits System, CAS 25(4), 208–214 (1976)
11. Hagen, L., Kahng, A.B.: A New Approach to Effective Circuit Clustering. In: IEEE International Conference on CAD, pp. 422–427 (1992)
12. Holland, J.H.: Adaption in Natural and Artificial Systems. University of Michigan, Press, Ann Arbor (1975)
13. Karger, P.G., Preas, B.T.: Automatic Placement: A Review of Current Techniques. In: Proceedings of The 23rd DAC, Las Vegas, Nevada, pp. 622–629 (1986)

14. Karypis, G., Aggarwal, R., Kumar, V., Shekhar, S.: Multilevel Hypergraph Partioning: Application in VLSI Design. In: Proceedings of DAC, Las Vegas, Nevada, pp. 526–529 (1997)
15. Kennings, A.: Cell Placement Using Constructive and Iterative Methods.: PhD thesis, University of Waterloo, Ont. Canada (1997)
16. Kernighan, B.W., Lin, S.: An Efficient Heuristic Procedure for Partitioning Graphs. The Bell System Technical Journal 49(2), 291–307 (1970)
17. Mallela, S., Grover, L.K.: Clustering Based Simulated Annealing for Standard Cell Placement. In: Proceedings of The 23rd DAC, Las Vegas, Nevada, pp. 312–317 (1989)
18. Schuler, D.M., Ulrich, E.: Clustering and Linear Placement. In: Proceedings of Design Automation Conference, Las Vegas, Nevada, pp. 50–56 (1972)
19. Shahookar, K., Mazumder, P.: VLSI Cell Placement Techniques. ACM Computing Surveys 23(2), 143–220 (1991)
20. Sun, W., Sechen, C.: Efficient and Effective Placement for Very Large Circuits. IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, 349–359 (1995)
21. Yang, Z., Areibi, S.: Global Placement Techniques for VLSI Physical Design Automation. In: 15th International Conference on Computer Applications in Industry and Engineering, San Diego, California (2002)

Bacterial Foraging Approach to Economic Load Dispatch Problem with Non Convex Cost Function

B. Padmanabhan[1], R.S. Sivakumar[2], J. Jasper[3], and T. Aruldoss Albert Victoire[4]

[1] Power systems Engineering
[2] Power systems Engineering
[3] Department of Electrical Engineering,
Anna University of Technology, Coimbatore
[4] Department of Electrical Engineering,
Anna University of Technology, Coimbatore

Abstract. This paper presents a bacterial foraging-based optimization (BFBO) technique to solve non-convex economic load dispatch (NCELD) problem of thermal plants. The presented methodology can take care of economic dispatch problems involving constraints such as transmission losses, valve point loading, ramp rate limits and prohibited operating zones. The idea of BFBO is motivated by the natural selection which tends to eliminate the animals with poor foraging strategies and favour those having successful foraging strategies. The BFBO method is tested with two power system cases consisting of 6 and 13 thermal units. Comparison with similar approaches including Genetic Algorithm (GA), particle swarm optimization (PSO) and other versions of differential evolution (DE) are given. The presented method outperforms other state-of-the-art algorithms in solving economic load dispatch problems with the valve-point effect.

Keywords: Bacterial foraging, Economic load dispatch, Non convex cost function, valve point effect.

1 Introduction

The objective of the economic load problem (ELD) is to schedule the committed generating unit outputs so as to meet the required load demand at minimum operating cost while satisfying all unit and system equality and inequality constraints [1].

In traditional EDPs, the cost function of each generator is approximately represented by a simple quadratic function and is solved using mathematical programming[2] based on several optimization techniques, such as dynamic programming [3], linear programming[4], homogenous linear programming[5], and nonlinear programming technique[6], [7]. However, real input–output characteristics display higher-order nonlinearities and discontinuities. Power plants usually have multiple valves that are used to control the power output of the unit. When steam admission valves in thermal units are first opened, a sudden increase in losses is observed. This leads to ripples in the cost function, which is known as the valve-point loading. The ELD problem with valve-point effects is represented as a non-smooth optimization [11].   .

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 577–584, 2011.
© Springer-Verlag Berlin Heidelberg 2011

The traditional algorithms can solve the ELD problems effectively only if the incremental fuel-cost curves of the generating units are monotonically increasing piece-wise linear functions. But, a practical ELD must include ramp rate limits, prohibited operating zones, valve-point effects and multi-fuel options. The resultant ELD is a challenging nonconvex optimization problem, which is hard to solve by the traditional methods. Recently, as an alternative to the conventional mathematical approaches, modern heuristic optimization techniques such as simulated annealing, evolutionary algorithms (EAs) (genetic algorithm, evolutionary programming, and evolution strategies) [8], particle swarm optimization, neural networks, and taboo search[9]-[12] have been given much attention by many researchers due to their ability to find an almost global optimal solution. These methods have drawbacks such as premature convergence and after some generations the population diversity would be greatly reduced.

Inspired from the mechanism of the survival of bacteria, e.g., E. coli, an optimization algorithm, called Bacterial Foraging Algorithm (BFA) [10], has been developed. Chemotaxis, reproduction and dispersion are the three processes with the help of which global searching capability of this algorithm has been achieved. These properties have helped BFA to be applied successfully in several kinds of power system optimization problems especially in solving the NCELD problem.

This paper considers two types of non-convex ELD problems, namely, ELD with prohibited operating zones and ramp rate limits (ELDPOZRR), ELD with valve-point loading effects (ELDVPL). The performance of the proposed method in terms of solution quality and computational efficiency has been compared with NPSO-LRS, CDE-QP and other techniques with non convex cost function.

2 Problem Formulation

The objective of the economic dispatch is to minimize the total generation cost of a power system over some appropriate period while satisfying various constraints.

The objective function can be formulated as,

$$F_T = \min \sum_{i \in \Psi} F_i(P_i) = \min \sum_{i \in \Psi} (a_i + b_i P_i + c_i P_i^2) \tag{1}$$

where F_T is the total fuel cost of all generating units, i is the index of dispatchable units; $F_i(P_i)$ is the cost function of the unit *i,* P_i is the power output of the unit *i*, Ψ is the set of all dispatchable units and a_i, b_i, c_i are the fuel cost coefficients of the unit *i*.The general PED problem consists in minimizing F_T subject to following constraints,

Power Balance Constraint:

$$\sum_{i \in \Psi} P_i - P_D - P_L = 0 \tag{2}$$

The transmission Loss P_L may be expressed using B-coefficients as,

$$P_L = \sum_{i\in\Psi}\sum_{j\in\Psi} P_i B_{ij} P_j + \sum_{i\in\Psi} B_{0i} P_i + B_{00} \tag{3}$$

where P_D is the total load demand; P_L is the power losses and B_{ij} is the power loss coefficient.

Generator Capacity Constraints:

The power generated by each unit lies within their lower limit $P_i^{\min}$ and upper limit $P_i^{\max}$. So that

$$P_i^{\min} \le P_i \le P_i^{\max} \tag{4}$$

2.1 ELDVPL

To consider the valve-point effects in the cost model, the rectified sinusoidal function should be incorporated into the quadratic function and the objective function min (F_T) is represented by a more complex formula along with (2), (3) and (4) as,

$$F_i(P_i) = a_i P_i^2 + b_i P_i + c_i + \left| e_i \sin(f_i (P_i^{\min} - P_i)) \right| \tag{5}$$

where e_i and f_i are the fuel cost coefficients of generator i reflecting valve point effects.

2.2 ELDRRPOZ

The operating range of units is restricted by their ramp rate limits during each dispatch period. Consequently the power output of a practical generator cannot be varied instantaneously beyond the range along with (2), (3), (4) and (5) as it is shown in the following expression:

$$\text{As generation increases, } P_i - P_{io} \le UR_i \tag{6}$$

$$\text{As generation decreases } P_{io} - P_i \le DR_i \tag{7}$$

and

$$\max\,(P_i^{\min}, P_{io} - DR_i) \le P_i \le \min(P_i^{\max}, P_{io} + UR_i) \tag{8}$$

Mathematically, the feasible operating zones of unit can be described as follows:

$$\begin{aligned} &P_{i\min} \le P_i \le P_{i,1}^{l} \ or \\ &P_{i,j-1}^{u} \le P_i \le P_{i,j}^{l}, j = 2,\dots,n_i \ or \\ &P_{i,n_i}^{u} \le P_i \le P_{i\max}, \forall i \in \psi \end{aligned} \tag{9}$$

where $P_{i,j}^{l}$ is the lower bound of the prohibited zone j of unit i, $P_{i,j}^{u}$ is the upper bound of the prohibited zone j of unit i, n_i be the number of prohibited zones in unit i, $P_{i\min}$ is the minimum generation limit of unit i and $P_{i\max}$ is the maximum generation limit of unit i.

3 Bacterial Foraging: Review

BFO method was invented by Kevin M. Passino motivated by the natural selection which tends to eliminate the animals with poor foraging strategies and favour those having successful foraging strategies. The foraging strategy are given

3.1 Chemotaxis

Chemotaxis process is the characteristics of movement of bacteria in search of food and consists of two processes namely swimming and tumbling. A bacterium is said to be 'swimming' if it moves in a predefined direction, and 'tumbling' if moving in an altogether different direction. Suppose $\theta^i(j,k,l)$ represents i th bacterium at j th chemotactic, k th reproductive and l th elimination dispersal step. $C(i)$ is the size of the step taken in the random direction specified by the tumble (run length unit). Then in computational chemotaxis the movement of the bacterium may be represented by

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \tag{10}$$

where Δ indicates a vector in the random direction whose elements lie in [–1, 1].

3.2 Swarming

It is always desired that the bacterium which has searched optimum path of food search should try to attract other bacteria so that they reach the desired that the bacterium which has searched optimum path of food search should try to attract other bacteria so that they reach the desired place more rapidly. Swarming makes the bacteria congregate into groups and hence move a concentric pattern of groups with high bacterial density. Mathematically, swarming can be represented by

$$J_{CC}(\theta, P(j,k,l)) = \sum_{i=1}^{S}[-d_{attract}\exp(-\omega_{attract}\sum_{m=1}^{p}(\theta_m-\theta_m^i)^2)] + \sum_{i=1}^{S}[h_{repellant}\exp(-\omega_{repellant}\sum_{m=1}^{p}(\theta_m-\theta_m^i)^2)] \tag{11}$$

where, J_{CC} is the penalty added to the original cost function. J_{CC} is basically the relative distances of each bacterium from the fittest bacterium. S is the number of bacterium, 'p' represents number of parameters to be optimized, θ_m is the position of the fittest bacterium, $d_{attract}$, $h_{repellant}$, $\omega_{attract}$ and $\omega_{repellant}$ are different coefficients.

3.3 Reproduction

In reproduction, population members who have had sufficient nutrients will reproduce and the least healthy bacteria will die. The healthier half of the population replaces with the other half of bacteria which gets eliminated, owing to their poorer foraging abilities. This makes the population of bacteria constant in the evolution process.

3.4 Elimination and Dispersal

A sudden unforeseen event may drastically alter the evolution and may cause the elimination and/or dispersion to a new environment. They have the effect of possibly destroying the chemotactic progress, but they also have the effect of assisting in chemotaxis, since dispersal may place bacteria near good food sources. Elimination and dispersal helps in reducing the behavior of stagnation i.e. being trapped in a premature solution point or local optima.

4 Adaptive Strategy

A variation to the BFBO is given by employing adaptive strategies to improve the performance of the BFBO which controls the exploration of the whole search space and the exploitation of the promising areas. This strategy is used to analyze the run length parameter of the BFBO. This improvement is achieved by enabling the bacterial foraging algorithm to adjust the run-length unit parameter dynamically during algorithm execution in order to balance the exploration/exploitation tradeoff. Living in groups allows individuals to allocate foraging effort between two different roles, namely, the producer and the scrounger. The "producer" can be used to locate food patches independently while the "scrounger" can be used to exploit the food discovered by other group members. The bacterial producers explore the search space and have the responsibility to find the promising domains and to leave the local optima that have been visited while the bacterial scroungers focus on the precision of the found solutions. The producer-scrounger foraging is used to dynamically determine the chemotactic step sizes for the whole bacterial colony during a run, hence dividing the foraging procedure of artificial bacteria colony into multiple explore and exploit phases.

4.1 Pseudocode

```
If (t mod n = 0) then
        If ( $f_{best} < \epsilon(t)$ ) then
                $C(t+1) = C(t-n) / \alpha$
                $\epsilon(t+1) = \epsilon(t) / \beta$
        ELSE
```

```
            C ( t+1 )=C (t-n);
            ϵ (t + 1)=ϵ (t - n);
        END IF
ELSE
C (t + 1)=C (t)
ϵ (t + 1)=ϵ (t);
END IF
```

where t is the current generation number, f_{best} is the best fitness value among all the bacteria in the colony, $\varepsilon(t)$ is the required precision in the current generation, and n, α, and β are user-defined constants.

5 Simulation Result

To validate the effectiveness of the proposed method, two test cases such as 6 unit and 13 unit systems with non convex cost function are taken. The result obtained from proposed method has been compared with CDE-QP [13] for 13 generator system; with MPSO and other techniques for 6 generator system. The software has been written in MATLAB-R2009a language and executed on a 2.0-GHz Pentium Dual personal computer with 1400-MB RAM.

Case Study 1

A system with six generators with ramp rate limit and prohibited operating zone is used here and has a total load of 1263 MW. The input data have been adopted from [13]. Results obtained from DE, BFBO, PSO and new coding-based modified PSO [12] and other methods have been presented here. Table 2 shows the frequency of convergence in 50 trial runs. It is clear from Table 1 shows that the proposed method produces a much better solution compared to the MPSO, NPSO-LRS, IDE and other methods.

Table 1. Comparison of cost among different methods for 50 trials (6-unit system)

Method	Generation Cost ($/h)		
	Maximum cost	Minimum cost	Average cost
GA	15,524	15,459	15,469
PSO	15,492	15,450	15,454
NPSO – LRS	15,452	15,450	15,450.5
MPSO	15,447	15,455	15,447
IDE	15,359	15,351	15,356
BFBO	15, 352	15,348	15,350

Table 2. Frequency of convergence for 50 trials (6-unit system)

Methods	Range of Generation Cost ($/h)		
	15000-15350	15350-15400	15400-15500
BFBO	32	10	8

Case Study 2

This test case is a NSELD of 13 units with valve point loading and has a load demand of 1800MW.The input data are given in [13]. The result obtained from presented method has been compared with CDE-QP [13] and other methods. Table 4 shows the frequency of convergence in 50 trial runs. It is clear from Table 3 and 4 that the proposed method produces a much better solution with less computation time compared to the ICA-PSO, CDE-QP and other methods.

Table 3. Comparison of cost among different methods for 50 trials (13-unit system)

Method	Generation Cost ($/h)		
	Maximum cost	Minimum cost	Average cost
PS	18404.04	18048.21	18190.32
STDE	18128.87	17963.89	18046.38
MDE	17969.09	17960.39	17967.19
ICA-PSO	17978.14	17960.37	17967.94
CDE-QP	17944.81	17938.95	17943.133
BFBO	17933.61	17918.73	17921.12

Table 4. Frequency of convergence for 50 trials (6-unit system)

Methods	Range of Generation Cost ($/h)		
	17800-17920	17920-17930	17930-17940
BFBO	28	12	10

6 Conclusion

This paper presents new combined approaches combining to solve the ELD problems of electric energy with the valve-point effect. The BFBO algorithm has the ability to find the better quality solution and has better convergence characteristics, computational efficiency, and robustness. It is clear from the results obtained by different trials that the proposed BFBO method has good convergence property and can avoid the shortcoming of premature convergence of other optimization techniques to obtain better quality solution. Two case studies have been used and the simulation results indicate that this optimization method is very accurate and converges very rapidly so that it can be used in the practical optimization problems. Due to these properties, the BFBO method in the future can be tried for solution of large unit systems and dynamic ELD problems in the search of better quality results.

References

[1] Abido, M.A.: A novel multiobjective evolutionary algorithm for environmental/ economical power dispatch. Elect. Power Syst. Res. 65, 71–81 (2003)

[2] Wood, A.J., Wollenberg, B.F.: Power Generation, Operation and Control. Wiley, New York (1994)

[3] Lin, C.E., Viviani, G.L.: Hierarchical economic dispatch for piecewise quadratic cost functions. IEEE Trans. Power App. Syst. PAS-103(6), 1170–1175 (1984)

[4] Yang, H.T., Chen, S.L.: Incorporating a multi-criteria decision procedure into the combined dynamic programming/production simulation algorithm for generation expansion planning. IEEE Trans. Power Syst. 4(1), 165–175 (1989)

[5] Granville, S.: Optimal reactive dispatch through interior point methods. In: Proc. IEEE Summer Meeting, Seattle, WA, Paper no. 92 SM 416-8 PWRS (1992)

[6] Liang, Z.X., Glover, J.D.: A zoom feature for a programming solution to economic dispatch including transmission losses. IEEE Trans. Power Syst. 7(3), 544–550 (1992)

[7] Chen, C.L., Wang, C.L.: Branch-and-bound scheduling for thermal generating units. IEEE Trans. Energy Convers. 8(2), 184–189 (1993)

[8] Park, J.-B., Lee, K.-S., Shin, J.-R., Lee, K.Y.: A particle swarm optimization for economic dispatch with nonsmooth cost function. IEEE Trans. Power Syst. 20(1), 34–42 (2005)

[9] Kumar, J., Seblé, G.B.: Clamped state solution of artificial neural network for real-time economic dispatch. IEEE Trans. Power Syst. 10(2), 925–931 (1995)

[10] Sinha, N., Chakrabarti, R., Chattopadhyay, P.K.: Evolutionary programming techniques for economic load dispatch. IEEE Trans. Evol. Comput. 7(1), 83–94 (2003)

[11] Victoire, T.A.A., Jeyakumar, A.E.: Hybrid PSO-SQP for economic dispatch with valve-point effect. Elect. Power Syst. Res. 71(1), 51–59 (2004)

[12] Walters, D.C., Sheble, G.B.: Genetic algorithm solution of economic dispatch with valve point loading. IEEE Trans. Power Syst. 8(3), 1325–1332 (1993)

[13] Panigrahi, B.K., Yadav, S.R., Agrawal, S., Tiwari, M.K.: A clonal algorithm to solve economic load dispatch. Electric Power System Research 77, 1381–1389 (2007)

Static/Dynamic Environmental Economic Dispatch Employing Chaotic Micro Bacterial Foraging Algorithm

Nicole Pandit[1], Anshul Tripathi[1], Shashikala Tapaswi[1], and Manjaree Pandit[2]

[1] ABV-IIITM, Gwalior, India
[2] Department of Electrical Engineering, MITS, Gwalior, India
{nicole0831,anshul.iiitm1}@gmail.com,
stapaswi@iiitm.ac.in,
manjaree_p@hotmail.com

Abstract. Environmental Economic Dispatch is carried out in the energy control center to find the optimal thermal generation schedule such that power balance criterion and unit operating limits are satisfied and the fuel cost as well as emission is minimized. Environmental economic dispatch presents a complex, dynamic, non-linear and discontinuous optimization problem for the power system operator. It is quite well known that gradient based methods cannot work for discontinuous or nonconvex functions as these functions are not continuously differentiable As a result, evolutionary methods are increasingly being proposed. This paper proposes a chaotic micro bacterial foraging algorithm (CMBFA) employing a time-varying chemotactic step size in micro BFA. The convergence characteristic, speed, and solution quality of CMBFA is found to be significantly better than classical BFA for a 3-unit system and the standard IEEE 30-bus test system.

1 Introduction

Bacterial Foraging Algorithm (BFA), proposed by Passino [1], is a new evolutionary optimization algorithm inspired by the social foraging behavior of *E. coli* bacteria found in human intestines. The biology behind the foraging strategy of *E. coli* is imitated and formulated into a powerful optimization algorithm. The bacteria search for nutrition with the help of two basic operations i.e. swim and tumble, together known as *chemotaxis.* However, the classical BFA is found to have certain limitations like slow convergence and saturation, particularly for large dimension problems. To handle these issues many improved versions of the classical BFA have been proposed. In reference [2] a micro-bacterial foraging algorithm is presented which evolves with a very small population. Ref [3] proposed a hybrid algorithm which combines the PSO with BFA to improve the efficiency and accuracy of the classical BFA.

Environmental economic dispatch (EED) is a highly non linear, discontinuous, non differentiable problem where objective function may have multiple local minima. Reference [4] summarizes the various algorithms for solving environmental dispatch problem. In [5] a method which combines cost and emission objectives into a single function using price penalty factor is proposed. The environmental economic dispatch problem has been solved using non dominated sorting GA [6], PSO based methods

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 585–592, 2011.
© Springer-Verlag Berlin Heidelberg 2011

[7] and bacterial foraging algorithm [8]. Most researchers have concentrated only on the static environmental economic dispatch problem (SEED) except ref. [6] where dynamic EED (DEED) has been implemented. This paper proposes a modified BFA by integrating a micro bacterial foraging algorithm [2] with chaotically varying chemotactic step size. The idea is to (i) reduce the time of operation by reducing bacteria count and (ii) to improve exploration and exploitation by employing a time varying step size. The results of the CMBFA are compared with the classical BFA and with results from literature, and are found to be superior.

2 Static/Dynamic Combined Environmental Economic Dispatch

The objective of static environmental economic dispatch (SEED) problem is to determine the generated powers P_i of units for a load of P_D so that the total fuel cost, F_T and total emission content E_T ,expressed below are simultaneously minimized.

$$F_T = \sum_{i=1}^{N} F_i(P_i) = \sum_{i=1}^{N} (a_i P_i^2 + b_i P_i + c_i) \$ / h \tag{1}$$

where a_i, b_i and c_i are the fuel-cost coefficients, $\alpha_i, \beta_i, \gamma_i, \xi_i$ are the pollution coefficients and P_i is the output of the i^{th}unit. Emission content of sulfur oxides (SO_x) and nitrogen oxides (NO_X) caused by fossil plants can be represented in ton/h as

$$E_T = \sum_{i=1}^{N} E_i(P_i) = \sum_{i=1}^{N} 10^{-2} * (\alpha_i + \beta_i P_i + \gamma_i P_i^2) + \xi_i \exp(\lambda_i P_i) \tag{2}$$

In combined EED formulation, the two objective problem is converted into a single objective problem using the price penalty factor, *pf* which blends the emission costs with the fuel cost and for i^{th} unit it is defined as F_i ($P_i^{\max}$)/ E_i ($P_i^{\max}$) [5]. The pf_i values are arranged in ascending order; the maximum capacity of each unit, $\left(P_i^{\max}\right)$ is added one at a time, starting from the smallest pf_i unit, until total demand P_D is met When the above condition is met, the pf_i associated with the last unit in the process is the price penalty factor *pf* ($/ton) for the given load P_D. Then equation (3) can be solved to obtain environmental economic dispatch.

Minimize

$$Z = \sum_{i=1}^{N} (a_i P_i^2 + b_i P_i + c_i) + pf * \sum_{i=1}^{N} (10^{-2} * (\alpha_i + \beta_i P_i + \gamma_i P_i^2)) + \xi_i \exp(\lambda_i P_i) \quad \$/\text{h} \tag{3}$$

Subject to Constraints

i) Unit operating limits constraint

$$P_i^{\min} \le P_i \le P_i^{\max} \qquad i = 1, 2, \ldots, N \tag{4}$$

ii) Power balance constraint

$$\sum_{i=1}^{N} P_i - \left(P_D + P_L\right) = 0 \tag{5}$$

The transmission losses are expressed as

$$P_L = \sum_{i=1}^{N} \sum_{j=1}^{N} P_i B_{ij} P_j + \sum_{i=1}^{N} B_{oi} P_i + B_{oo} \tag{6}$$

iii) Unit ramp-rate limit constraints

When the generator ramp rate limits are considered, the operating limits are modified as follows:

$$Max(P_i^{\min}, P_i^{o} - DR_i) \le P_i \le Min(P_i^{\max}, P_i^{o} + UR_i) \tag{7}$$

The previous operating point of ith generator is P_i^o and DR_i and UR_i are the down and up ramp rate limits respectively.

iv) Prohibited operating zones

The cost curves of practical generators are discontinuous as whole of the unit operating range is not always available for allocation and can be expressed as follows:

$$P_i \in \begin{cases} P_i^{\min} \le P_i \le P_{i1}^{L} \\ P_{ik-1}^{U} \le P_i \le P_{ik}^{L} \\ \text{..........} \quad \text{..........} \quad \ldots \\ P_{izi}^{U} \le P_i \le P_i^{\max} \end{cases} \tag{8}$$

Here z_i are the number of prohibited zones in ith generator curve, P_{ik}^{L} is the lower limit and P_{ik-1}^{U} is the upper limit of kth prohibited zone of ith generator.

2.1 Dynamic Economic Emission Dispatch

Dynamic economic emission dispatch (DEED) deals with sharing the system load including system losses among the available generators in such a way that all equality and inequality constraints mentioned above are met and the cost as well as emission are minimized for each time interval.

The combined dynamic economic emission dispatch (DEED) problem is formulated for each time interval as defined in eq. (3).

3 Classical BFA Algorithm

The bacterial foraging algorithm (BFA) was introduced by Kevin M. Passino [1] motivated by the group foraging strategy of a swarm of *E. coli* bacteria which search for nutrients so as to maximize energy obtained per unit time. The process, by which a bacterium moves by taking small steps while searching for nutrients, is called chemotaxis. The main idea of BFA is imitating chemotactic movement of virtual bacteria in the problem search space. The classical BFA algorithm consists of four main steps, wiz. *chemotaxis*, *swarming, reproduction*, and *elimination and dispersal.*

The movement of the bacterium may be represented as:

$$\theta^i(j+1,k,l)=\theta^i(j,k,l)+C(i)\phi(j) \text{ where } \phi(j)=\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \tag{9}$$

Where $\theta_i(j,k,l)$ is the position of the i^{th} bacterium at j^{th} chemotactic k^{th} reproductive, and l^{th} elimination and dispersal step. C(i) is the size of the step taken in the random direction of tumble. Here, Δ indicates a movement in arbitrary direction whose elements lie in [-1, 1].

3.1 Chaotic Micro Bacterial Foraging Algorithm (CMBFA)

In micro BFA [2] a 3-bacteria population searches iteratively for large dimension complex domain. A chaotically varying chemotactic step size is employed to update the position of the bacteria. The best bacterium retains its position. The second best bacterium is moved to a position very close to the best one, the least fit bacterium is killed and its place is taken by a new bacterium. Reproduction is not carried out as the bacteria population is very small. Elimination dispersion step of original algorithm is also retained. The step size C in eq.(9) is changed chaotically making use of one of the dynamic systems called the logistic map.

$$C(t)=\mu\times C(t-1)\times[1-C(t-1)] \tag{10}$$

Here, t is the iteration count and μ is a parameter, $0 \le \mu \le 4$ which controls the variation of the chaotic sequence. The step size C is iteratively varied. The system at (10) displays chaotic behavior when μ=4 and $C(0)\notin\{0,0.25,0.5,0.75,1.0\}$. Fig. 1

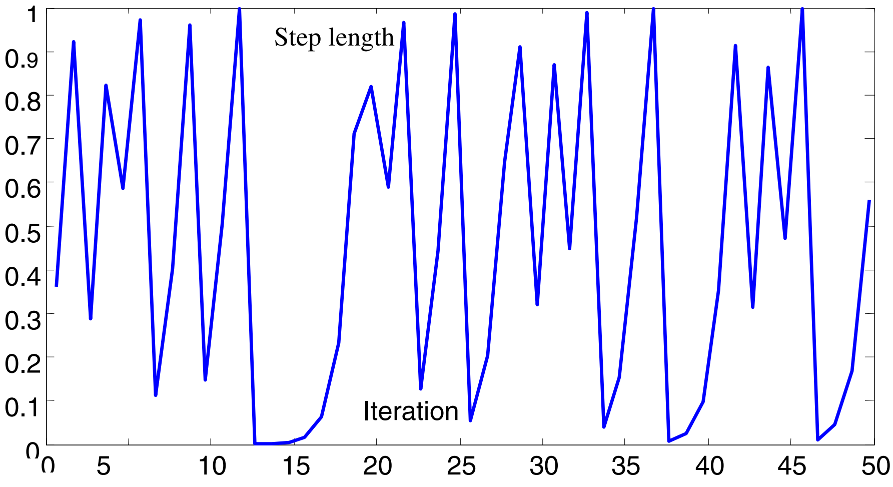


Fig. 1. Step length in CMBFA for μ =4, C*(t=0)* =0.64

shows the variation of step length for C(0)=0.64. For C(0) =(0.0, 0.25,0.5,0.75, 1.0) there will be no variation in step length with time..

4 Results and Discussion

The performance of CMBFA is compared for test systems having i) Prohibited operating zones(POZ) ii) ramp rate limits (RRL)and iii) transmission losses in addition to the iv)generating capacity constraints and v) power balance equation. The optimal solutions are computed for i) Test case I: SED for three thermal generating units system [9] with POZ and RRL without losses ii) Test Case II: Test case I with transmission losses iii) Test Case III: IEEE 30-bus test system for SEED problem [7]. iv)Test Case IV: The DEED for Test Case III for 24-hour load variation. Simulations were carried out using MATLAB 7.0.1 on a Pentium IV processor, 2.8 GHz. with 1 GB RAM. The RRL is assumed as [.08,.11,.15,.18,.15,.18].

4.1 Parameter Setup and Effect of Bacteria Population

For both BFA and CMBFA, the number of chemotactic steps is taken as 50, length of swimming loop and elimination dispersal loop are both set at 4 and number of iterations are 100. For the Test case I base load was 300 MW. Results for different bacteria counts are tabulated in Table 1. The standard deviation (S.D) improves with bacteria count.

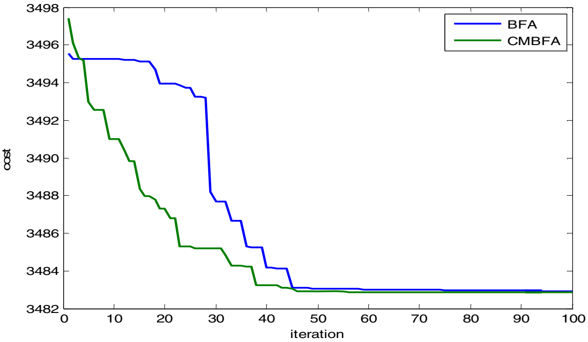


Fig. 2. Comparison of convergence characteristics of BFA and CMBFA for test case I

Table 1. Effect of change in bacteria count on BFA performance: Test Case I (out of ten trials)

Bacteria count	Minimum Cost($/hr)	Maximum Cost($/hr)	Mean	SD
4	3482.86797	3484.91036	3483.1	0.3666
6	3482.86749	3482.87451	3482.9	0.0020
8	3482.86798	3482.88374	3482.9	0.0013
10	3482.86796	3484.90744	3483.1	0.00046
12	3482.86760	3482.86970	3482.9	0.00026

The CPU time required increases with bacteria count. The convergence behavior of BFA and CMBFA is plotted in Fig. 2. The CMBFA converges faster with CPU time 0.1235s/iteration as compared to BFA with 0.2454s/iteration.

The performance of CMBFA is compared with GA[9] APSO[10], AIS[11] and HBBO[12] in Table 2. The results of CMBFA are superior to others as minimum cost at zero constraint violation is achieved. In [10] and [12] the ramp rate limits for unit three are violated and in [9] and [11] the constraint violation (shown bold) is high which makes the result infeasible.

Table 2. Comparison of best results for Test Case II

Output(MW)	CMBFA	BFA	HBBO[12]	APSO[10]	AIS[11]	GA[9]
P1	200.4637	205.88944	207.637	200.528	198.7575	194.26
P2	78.3750	73.06689	87.2833	78.2776	77.99225	50
P3	34.0000	34.0000	**15.0000**	**33.9918**	35.19886	79.62
Total power	312.8387	312.9563	309.9203	312.7974	311.9486	323.88
loss	12.8387	12.9572	9.9204	12.8364	11.816	24.011
$\sum P_i - P_D - P_L$	0.0	-0.0009	-0.0001	-0.039	**0.1326**	**-0.131**
cost ($/h)	3634.7691	3635.1583	3619.7565	3634.3127	3624.44	3737.20

4.2 Static/Dynamic Environmental Economic Dispatch (SEED/DEED)

Table 3 compares BFA and CMBFA for best cost and best emission cases. The results of CMBFA are significantly better for both cases. To get the best compromise solution, the price penalty factor *pf* helps in obtaining a solution which gives equal weight to minimization of cost as well as emission. In Table 4, for Load 2.4 MW to 2.8 MW, pf=4470.2746 and for larger values of Load, pf=5928.7134.

Table 3. Comparison of best cost and emission solutions by BFA and CMBFA for Test Case III

Variables	BEST COST SOLUTION		BESTEMISSION SOLUTION	
	BFA	CMBFA	BFA	CMBFA
P1(per unit)	0.10422553	0.1209	0.40409088	0.3953
P2 (per unit)	0.28841535	0.2812	0.38769951	0.4304
P3 (per unit)	0.58869564	0.6050	0.48109197	0.5386
P4 (per unit)	0.94961114	0.9801	0.46402481	0.4238
P5 (per unit)	0.54599832	0.5239	0.65923582	0.5584
P6 (per unit)	0.38274581	0.3473	0.47191157	0.5214
COST(&/h)	606.34386659	605.9438	637.61844944	641.5610
EMISSION(ton/h)	0.22571781	0.2484	0.19512784	0.1937
LOSS(MW/h)	0.02540773	0.0248	0.03407091	0.0339
$\sum P_i - P_D - P_L$	0.00028407	-0.0005	-0.00001635	-0.0000

Table 4. Best Compromise Solution for DEED using CMBFA for IEEE 30-bus system

Hou	P1	P2	P3	P4	P5	P6	Cost	Emission	LOSS	LOAD
H1	0.1195	0.5013	0.5483	0.4284	0.3586	0.4682	537.0861	0.2014	0.0242	2.4
H2	0.2195	0.4582	0.5483	0.4930	0.3586	0.3946	541.6875	0.1993	0.0222	2.45
H3	0.2195	0.3613	0.3683	0.6284	0.5386	0.4078	543.1863	0.2018	0.0240	2.5
H4	0.2195	0.3613	0.3683	0.5149	0.5386	0.5946	570.7961	0.1997	0.0273	2.57
H5	0.1195	0.5013	0.5483	0.5332	0.5386	0.3946	575.0908	0.2014	0.0256	2.61
H6	0.2195	0.5013	0.3683	0.6284	0.4624	0.5115	586.9743	0.2004	0.0313	2.66
H7	0.2195	0.5013	0.4559	0.6284	0.5386	0.3946	594.2952	0.1999	0.0283	2.71
H8	0.2195	0.5013	0.5483	0.6284	0.4953	0.3946	605.0353	0.1997	0.0273	2.76
H9	0.2165	0.5013	0.5483	0.6284	0.5386	0.3946	613.9105	0.1996	0.0278	2.8
H10	0.2123	0.4881	0.5052	0.5288	0.5386	0.5920	632.0021	0.1975	0.0311	2.834
H11	0.2195	0.4000	0.5483	0.6284	0.5386	0.5946	638.2734	0.2000	0.0293	2.9
H12	0.2195	0.5013	0.5483	0.6284	0.5386	0.5256	646.3690	0.2169	0.0313	2.93
H13	0.2195	0.5013	0.5483	0.5498	0.5386	0.5946	651.0041	0.1977	0.0321	2.92
H14	0.2195	0.4706	0.5483	0.4921	0.5386	0.5946	633.4414	0.1974	0.0298	2.834
H15	0.2195	0.5013	0.4200	0.5950	0.5132	0.5840	622.1056	0.1996	0.0332	2.8
H16	0.2195	0.5013	0.5279	0.4284	0.5386	0.5946	627.1872	0.1966	0.0303	2.78
H17	0.2195	0.5013	0.4460	0.6284	0.5386	0.3946	592.1582	0.2003	0.0284	2.7
H18	0.2195	0.5013	0.5483	0.6284	0.3942	0.3946	583.2600	0.2012	0.0262	2.66
H19	0.2195	0.3613	0.4250	0.5459	0.5386	0.5452	574.8516	0.2003	0.0257	2.61
H20	0.2195	0.4623	0.4583	0.4799	0.4499	0.5266	572.2063	0.1984	0.0266	2.57
H21	0.2195	0.3613	0.5483	0.4284	0.5386	0.4240	553.2874	0.1977	0.0202	2.5
H22	0.2195	0.3613	0.5271	0.4284	0.5386	0.3946	541.7397	0.1979	0.0196	2.45
H23	0.2195	0.3993	0.4255	0.4933	0.4307	0.4644	531.6621	0.1984	0.0226	2.41
H24	0.2195	0.3613	0.3683	0.5697	0.3586	0.4854	512.8081	0.2018	0.0227	2.34

5 Conclusions

A chaotic micro BFA (CMBFA) is developed in this paper and its performance is compared with the classical BFA for four test cases having complex, discontinuous and non-linear objective functions and constraints. The test results clearly show that i) the consistency of classical BFA depends on bacteria count. The mean cost and S.D improved with bacteria count. ii) The CMBFA is found to be many times faster.iii) The CMBFA has got a superior performance as compared to classical BFA. iv) BFA and CMBFA both produce feasible solutions with full constraint satisfaction but CMBFA produced lower cost, emission and losses as compared to BFA.

References

1. Passino, K.M.: Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Syst. Mag. 22(3), 52–67 (2002)
2. Dasgupta, S., Biswas, A., Das, S., Panigrahi, B.K., Abraham, A.: A Micro-Bacterial Foraging Algorithm for High-Dimensional Optimization. In: IEEE Congress on Evolutionary Computation, pp. 785–792 (2009)
3. Long, L.X., Jun, L.R., Ping, Y.: Bacterial Foraging Global Optimization Algorithm Based On the Particle Swarm Optimization. In: IEEE International Conference on Intelligent Computing and Intelligent Systems, pp. 22–27 (2010)

4. Talaq, J.H., El-Hawary, F., El-Hawary, M.E.: A summary of environmental and economic dispatch algorithms. IEEE Trans. Power Syst. 9(3), 1508–1516 (1994)
5. Palanichamy, C., Srikrishna, K.: Economic Thermal Power Dispatch with Emission Constraint. Journal of the Institution of Engineers (India) 72, 11–18 (1991)
6. Basu, M.: Dynamic economic emission dispatch using nondominated sorting genetic algorithm-II. Electrical Power and Energy Systems 30, 140–149 (2008)
7. Abido, M.A.: Multiobjective particle swarm optimization for environmental/economic dispatch problem. Electric Power Systems Research 79, 1105–1113 (2009)
8. Hota, P.K., Barisal, A.K., Chakrabarti, R.: Economic emission load dispatch through fuzzy based bacterial foraging algorithm. Electrical Power and Energy Systems 32, 794–803 (2010)
9. Chen, P.H., Chang, H.C.: Large scale economic dispatch approach by genetic algorithm. IEEE Transactions on Power Systems 10(4), 1919–1926 (1995)
10. Panigrah, B.K., Ravikumar, P.V., Das, S.: An Adaptive Particle Swarm Optimization Approach for Static and Dynamic Economic Load Dispatch. International Journal on Energy Conversion and Management 49, 1407–1415 (2008)
11. Panigrahi, B.K., Yadav, S.R., Agrawal, S., Tiwari, M.K.: A clonal algorithm to solve economic load dispatch. Electric Power System Research 77, 1381–1389 (2007)
12. Bhattacharya, A., Chattopadhyay, P.K.: Hybrid differential evolution with biogeography-based optimization for solution of economic load dispatch. IEEE Transactions on Power System 25(4), 1955–1964 (2010)

Artificial Bee Colony Algorithm with Self Adaptive Colony Size

Tarun Kumar Sharma[1], Millie Pant[1], and V.P. Singh[2]

[1] IIT Roorkee
[2] MIT Saharanpur
{taruniitr1,millidma,singhvp3}@gmail.com

Abstract. The Artificial Bee Colony or ABC is one of the newest additions to the class of Nature Inspired Algorithms (NIA) mimicking the foraging behavior of honey bees. In ABC, the food locations represent the potential candidate solution. In the present study an attempt is made to generate the population of food sources (Colony Size) adaptively with the help of proposed algorithm ABC-SAC (artificial bee colony with self-adaptive colony). ABC-SAC is further is modified by varying the behavior of bees in search of food and the corresponding variants are named as ABC-SAC1 and ABC-SAC2. The proposed algorithms are validated on a set of standard benchmark problems with varying dimensions taken from literature and the numerical results are compared with the basic ABC and one its recent variant gbest ABC, which indicate the competence of the proposed algorithms.

Keywords: Artificial Bee Colony, Self Adaptive, Foraging behaviour, Optimization.

1 Introduction

Artificial Bee Colony (ABC) optimization algorithm is a population-based swarm intelligence algorithm that was originally proposed by Karaboga, Erciyes University of Turkey in 2005 [1] [2]. It simulates the foraging behaviour that a swarm of bees perform. In this algorithm there are three groups of bees, the employed bees (bees that determines the food source (possible solutions) from a prespecified set of food sources and share this information (waggle dance) with the other bees in the hive), the onlookers bees (gets the information of food sources from the employed bees in the hive and select one of the food source to gathers the nectar) and the scout bees (responsible for finding new food sources). A brief overview of the algorithm is given in section 2. Like other population based search algorithm, ABC starts with a population of potential candidate solutions which in case of ABC represent the food locations (flower patch etc.). The amount of nectar in the food source represents its goodness (fitness value). In the basic ABC, the number of food sources is fixed in the beginning of the algorithm. However, practically speaking it is very rare that the every patch will contain the same number of flowers. Keeping this in mind, in the present study we propose a modified variant of ABC in which the population of food sources change adaptively. The corresponding algorithm is named ABC-SAC which

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 593–600, 2011.
© Springer-Verlag Berlin Heidelberg 2011

is further modified into two variants ABC-SAC1 and ABC-SAC2. In ABC-SAC1, elitism is used where the bees are always guided towards the best food source (i.e. the one having the best fitness function value). In ABC-SAC2, both local and global explorations are kept into account. The remaining of the paper is organized as follows; in the next section we give a brief overview of the basic ABC algorithm. In section 3, the proposed variants are described. Numerical results are given in section 4 and finally the paper concludes with section 5.

2 Artificial Bee Colony Algorithm

ABC is one of the newest algorithms based on the foraging behavior of insects. It tries to model natural behavior of real honey bees in food foraging. Since ABC algorithm is simple in concept, easy to implement, and has fewer control parameters, it has been widely used in many fields. ABC algorithm has been applied successfully to a large number of various optimization problems [3-10]. The colony of artificial bees contains three groups of bees: employed bees, onlookers and scouts. A bee waiting on the dance area for making a decision to choose a food source is called onlooker and one going to the food source visited by it before is named employed bee. The other kind of bee is scout bee that carries out random search for discovering new sources. The position of a food source represents a possible solution to the optimization problem and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution. In the algorithm, the first half of the colony consists of employed artificial bees and the second half constitutes the onlookers. The number of the employed bees or the onlooker bees is equal to the number of solutions in the population. At the first step, the ABC generates a randomly distributed initial population of NP solutions (food source positions), where NP denotes the size of population. Each solution xi where i =1, 2,..., NP is a D-dimensional vector, where D is the number of optimization parameters. After initialization, the population of the positions (solutions) is subjected to repeated cycles, C =1, 2,..., MCN of the search processes of the employed bees, the onlooker bees and scout bees. An employed bee produces a modification on the position (solution) in her memory depending on the local information (visual information) and tests the nectar amount (fitness value) of the new source (new solution). Provided that the nectar amount of the new one is higher than that of the previous one, the bee memorizes the new position and forgets the old one. Otherwise she keeps the position of the previous one in her memory. After all employed bees complete the search process; they share the nectar information of the food sources and their position information with the onlooker bees on the dance area. An onlooker bee evaluates the nectar information taken from all employed bees and chooses a food source with a probability related to its nectar amount. As in the case of the employed bee, she produces a modification on the position in her memory and checks the nectar amount of the candidate source. Providing that its nectar is higher than that of the previous one, the bee memorizes the new position and forgets the old one. An artificial onlooker bee chooses a food source depending on the probability value associated with that food source pi, calculated as Eq. (1):

$$p_i = \frac{fit_i}{\sum_{i=1}^{NP} fit_i} \tag{1}$$

where *fit*$_i$ is the fitness value of the solution i which is proportional to the nectar amount of the food source in the position i and NP is the number of food sources which is equal to the number of employed bees. In order to produce a candidate food position from the old one in memory, the ABC uses the following Eq. (2):

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \tag{2}$$

where k∈{1, 2, …, NP} and j∈{1, 2, …, D} are randomly chosen indexes. Moreover, k ≠ i. $\O_{ij}$ is a random number between [-1, 1]. It controls the production of neighbor food sources around xij and represents the comparison of two food positions visible to a bee. This can be seen from Eq. (2), as the difference between the parameters of the xij and xkj decreases, the perturbation on the position xij decreases, too. Thus, as the search approaches to the optimum solution in the search space, the step length is adaptively reduced. After each candidate source position is produced and evaluated by the artificial bee, its performance is compared with that of its old one. If the new food source has equal or better quality than the old source, the old one is replaced by the new one. Otherwise, the old one is retained. If a position cannot be improved further through a predetermined named “limit”, then that food source is assumed to be abandoned. The corresponding employed bee becomes a scout. The abandoned position will be replaced with a new food source found by the scout. Assume that the abandoned source xi, then the scout discovers a new food source to be replaced with xi. This operation can be defined as in Eq. (3):

$$x_i^j = x_{\min}^j + rand()(x_{\max}^j - x_{\min}^j) \tag{3}$$

where $x_{\max}^j$ and $x_{\min}^j$ are upper and lower bounds of parameter *j*, respectively.

3 Proposed ABC-SAC Algorithms and Its Variant

In the proposed ABC-SAC algorithm an attempt is to generate adaptive Colony Size. First of all we randomly generate some initial population of solutions, which at a later stage keep on changing adaptively. The Food Source position (NP) of subsequent generations is taken as the average of the population size attribute from all individuals in the current population as follows:

```
Initialize the population of solutions xi,j,G. Then
declare p as variable (Initialize p=0)

for (i=0; i<FoodNumber; i++)

int p = p +  Foods[i][D];

FoodNumber = (int) (p/FoodNumber) + 0.5);
```

Where FoodNumber is the Colony Size (The number of food sources equals the half of the colony size) and Foods is the population of food sources. (Check population size for even as number of food sources equals the half of the colony size)

It can be observed from the above code that the population of food source *(NP)* depend on p and may vary in every generation. The only thing to be kept in mind is that the population size should be even in number as the number of the employed bees or the onlooker bees is equal to the number of solutions in the population. This is a profitable situation as it may reduce the number of function evaluations leading to a faster convergence. The population generated in each iteration, using the above code, is shown in Fig. 1.

ABC-SAC1: The proposed ABC-SAC is further modified by including in it a concept of elitism, due to which the bees are always guided towards the best food source. Here, the food source is generated as follows:

$$v_{i,j} = x_{best,j} + \phi_{ij}(x_{r1,j} - x_{k,j})$$

where $x_{\text{best,j}}$ indicates the best food location. The remaining notations have the same meaning as defined in the previous section. This particular modification will further aid in getting a faster convergence. The bees always look for the best solution.

ABC-SAC2: In the other variant of the proposed ABC-SAC2 algorithm is further implemented to improve the exploitation of ABC by using the equation given below:

$$v_{i,j} = x_{i,j} + \phi_{ij}(x_{i,j} - x_{k,j}) + C(x_{best,j} - x_{r1,j})$$

where C is taken as 1.5 and r_1 is rand number between [0,1] .

This variant is inspired from the Particle Swarm Optimization (PSO) [11] (though the equation is not exactly same) which takes care of global as well as local exploration so that the domain of the problem is thoroughly is explored.

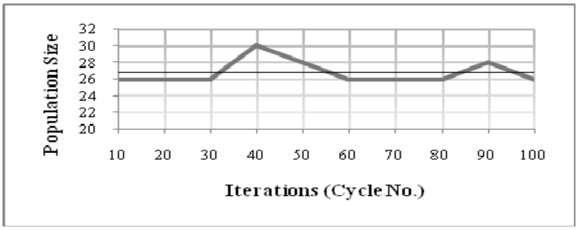


Fig. 1. Graph showing Colony Size (Population Size) (Dark line shows the adaptive colony)

4 Parameters Settings and Comparison Criterion

The ABC has certain control parameters that have to be set by the user. First of all it's the population (number) of food sources (colony size) which is taken as 100 for ABC and for ABC-SAC and its variants ABC-SAC1 and ABC-SAC2. The percentage of onlooker bees is 50% of the colony, the employed bees are 50% of the colony and the number of scout bees is selected as one for ABC. The *limit* (or generation) is taken 100. Random numbers are generated using inbuilt function *rand()* in DEVC++.

Besides comparing ABC-SAC and its variants ABC-SAC1 and ABC-SAC2 with basic ABC, we have also compared it with a latest variant of ABC called gbest ABC [11] in terms of best and mean fitness function values, standard deviation (SD) and average number of function evaluations (NFE). It should be noted here that according to literature in most of the ABC algorithms the stopping criteria is that of the number of cycles (or generations) however, in the present study we have considered NFE as a stopping criteria. The tests are repeated for 30 runs and the average number of function evaluations (NFE) is recorded. The maximum number of function evaluations is set 10^6. In every case, a run was terminated when an accuracy of $| f_{\max} - f_{\min} | < 10^{-20}$ was reached or when the maximum number of function evaluation was reached. Values lesser than 10^{-20} are treated as Zero (*0*). All algorithms are executed on Pentium IV, using DEV C++.

5 Benchmark Problems, Results and Discussion

The proposed ABC-SAC and its variants are validated on a set of 5 standard, scalable benchmark problems for different dimensions. The test problems are selected from [12]. The first four problems (Sphere, Griewank, Rastringin and Ackley) are tested for dimensions 30 and 60, while the remaining function (Schaffer) is tested for dimensions 2 and 3. The corresponding results are recorded in Tables 1 to 5. From these Tables we can see that all the proposed variants performed quite well in terms of NFE and fitness value. Among the proposed versions, ABC-SAC1 emerged as a clear winner outperforming the other algorithms and the remaining variants significantly. The performance curves of selected benchmark problems all the algorithms are

Table 1. Best, mean, standard deviation (std) and number of functions evaluations (NFE) for Sphere Function

	D = 30			
Algorithm	Best	Mean	SD	NFE
ABC	5.26E-16	6.58E-16	1.32E-16	33620
ABC - SAC	7.62E-18	8.94E-18	6.82E-18	27413
ABC – SAC1	1.70E-19	2.54E-19	3.25E-18	23191
ABC - SAC.2	3.30E-18	4.23E-18	1.91E-18	25672
gBest (C =1.5)	2.04E-16	4.18E-16	7.36E-17	31312
	D = 60			
Algorithm	Best	Mean	SD	NFE
ABC	2.90E-15	3.12E-15	5.16E-15	45650
ABC – SAC	4.23E-17	6.72E-17	1.05E-17	42124
ABC – SAC1	3.44E-18	4.29E-18	1.33E-18	39766
ABC - SAC.2	2.34E-18	3.57E-18	2.81E-18	41567
gBest (C =1.5)	1.56E-16	1.43E-15	1.38E-16	43544

Table 2. Best, mean, standard deviation (std) and number of functions evaluations (NFE) for Griekwank Function

	D = 30			
Algorithm	Best	Mean	SD	NFE
ABC	3.31E-16	4.82E-16	8.67E-17	47780
ABC - SAC	3.82E-18	5.07E-18	9.19E-18	40848
ABC – SAC1	4.83E-19	5.04E-19	8.31E-19	37905
ABC - SAC.2	5.62E-18	6.40E-18	9.73E-18	40131
gBest (C =1.5)	1.01E-17	2.96E-17	4.99E-17	42831

	D = 60			
Algorithm	Best	Mean	SD	NFE
ABC	7.59E-15	9.54E-15	7.15E-16	57975
ABC - SAC	8.65E-18	9.30E-18	4.80E-18	51695
ABC – SAC1	7.39E-19	8.40E-19	1.24E-19	43658
ABC - SAC.2	6.35E-18	9.00E-18	9.62E-18	40252
gBest (C =1.5)	6.97E-16	7.55E-16	4.13E-16	51947

Table 3. Best, mean, standard deviation (std) and number of functions evaluations (NFE) for Rastringin Function

	D = 30			
Algorithm	Best	Mean	SD	NFE
ABC	1.38E-13	2.19E-13	1.81E-13	74867
ABC - SAC	7.98E-15	7.98E-15	1.69E-15	68002
ABC – SAC1	5.39E-16	6.79E-16	1.40E-17	61040
ABC - SAC.2	6.68E-15	8.87E-15	9.13E-16	65200
gBest (C =1.5)	1.08E-14	1.33E-14	2.45E-14	71280

	D = 60			
Algorithm	Best	Mean	SD	NFE
ABC	2.29E-13	6.31E-13	2.63E-13	99342
ABC - SAC	1.16E-15	4.20E-15	2.22E-16	73912
ABC – SAC1	1.56E-16	5.01E-16	2.74E-16	67402
ABC - SAC.2	6.45E-16	8.43E-16	9.52E-17	71035
gBest (C =1.5)	1.40E-13	3.52E-13	1.24E-13	81430

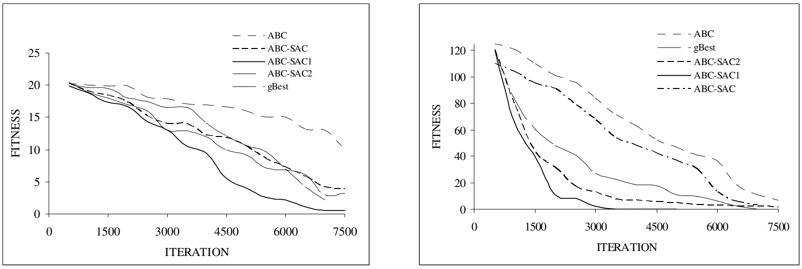
illustrated in Fig. 2. These figures clearly indicate the faster convergence of ABC-SAC. The faster convergence of ABC-SAC1 is due to the presence of elitism which always guides the bees towards the best food source. This increases the convergence rate and results in smaller NFE.

Table 4. Best, mean, standard deviation (std) and number of functions evaluations (NFE) for Ackley Function

	D = 30			
Algorithm	Best	Mean	SD	NFE
ABC	1.01E-13	1.11E-13	1.22E-13	97140
ABC - SAC	1.24E-14	2.01E-14	1.82E-14	83720
ABC – SAC1	2.61E-15	4.44E-15	3.57E-15	76900
ABC - SAC.2	9.02E-15	1.88E-14	2.23E-14	86620
gBest (C =1.5)	2.02E-14	3.22E-14	3.25E-15	91404
	D = 60			
Algorithm	Best	Mean	SD	NFE
ABC	8.64E-13	9.18E-13	4.36E-13	109090
ABC - SAC	5.28E-14	7.45E-14	1.94E-14	91834
ABC – SAC1	8.36E-15	9.39E-15	4.84E-16	83195
ABC - SAC.2	7.77E-14	9.11E-14	7.64E-14	88039
gBest (C =1.5)	2.47E-14	1.00E-13	6.09E-15	95301

Table 5. Best, mean, standard deviation (std) and number of functions evaluations (NFE) for Schaffer Function

	D = 2			
Algorithm	Best	Mean	SD	NFE
ABC	0	0	0	48760
ABC - SAC	0	0	0	41820
ABC – SAC1	0	0	0	32992
ABC - SAC.2	0	0	0	33080
gBest (C =1.5)	0	0	0	44500
	D = 3			
Algorithm	Best	Mean	SD	NFE
ABC	2.18E-16	5 .76E-16	1.62E-17	59900
ABC - SAC	7.78E-18	9.48E-18	6.30E-18	47450
ABC – SAC1	2.03E-19	7.56E-19	2.39E-19	39530
ABC - SAC.2	2.03E-06	6.51E-18	2.86E-19	35720
gBest (C =1.5)	1.01E-19	1.85E-18	1.01E-17	52502



(a) (b)

Fig. 2. Convergence Graph of (a) Ackley and (b) Sphere function

6 Conclusion

In the proposed study, we presented the concept of adaptive population of food sources for an Artificial Bee Colony algorithm. The corresponding algorithm named ABC-SAC was further modified by including elitism (ABC-SAC1) and by incorporating global-local exploration (ABC-SAC2) in it. It was observed that by self adapting the food sources, the performance of the basic ABC can be improved. It was also analyzed that this modification can be further improved by considering elitism as the proposed ABC-SAC1 showed better results in comparison to other variants. However, considering the narrow structure of the test bed we cannot make any concrete judgment about the performance the proposed ABC variants but by considering the results we may say that the proposed approach looks promising.

References

1. Karaboga, D., Basturk, B.: A Powerful And Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm. Journal of Global Optimization 39(3), 459–471 (2007)
2. Karaboga, D., Basturk, B.: On the performance of artificial bee colony (ABC) algorithm. Applied Soft Computing 8, 687–697 (2008)
3. Singh, A.: An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem. Applied Soft Computing 9, 625–631 (2009)
4. Karaboga, N.: A new design method based on artificial bee colony algorithm for digital IIR filters. Journal of The Franklin Institute 346, 328–348 (2009)
5. Ponton, J.W., Klemes, J.: Alternatives to neural networks for inferential measurement. Computers and Chemical Engineering 17, 42–47 (1993)
6. Rao, R.S., Narasimham, S., Ramalingaraju, M.: Optimization of distribution network conFigureuration for loss reduction using artificial bee colony algorithm. International Journal of Electrical Power and Energy Systems Engineering (IJEPESE) 1, 116–122 (2008)
7. Karaboga, D., Akay, B., Ozturk, C.: Artificial Bee Colony (ABC) Optimization Algorithm for Training Feed-Forward Neural Networks. In: Torra, V., Narukawa, Y., Yoshida, Y. (eds.) MDAI 2007. LNCS (LNAI), vol. 4617, pp. 318–329. Springer, Heidelberg (2007)
8. Venkata Rao, R., Pawar, P.J.: Parameter optimization of a multi-pass milling process using non-traditional optimization algorithms. Applied Soft Computing 10(2), 445–456 (2008)
9. Karaboga, D., Basturk, B.: Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems. In: Melin, P., Castillo, O., Aguilar, L.T., Kacprzyk, J., Pedrycz, W. (eds.) IFSA 2007. LNCS (LNAI), vol. 4529, pp. 789–798. Springer, Heidelberg (2007)
10. Pan, Q.-K., Tasgetiren, M.F., Suganthan, P.N., Chua, T.J.: A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem. Information Sciences (in press)
11. Zhu, G., Kwong, S.: Gbest-Guided Artificial Bee Colony Algorithm for Numerical Function Optimization. Appl. Math. Comput. (2010)
12. Fan, H.Y., Lampinen, J., Dulikravich, G.S.: Improvements to mutation donor formulation of differential evolution. International Congress on Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems Eurogen (2003)

Multi-Robot Box-Pushing Using Non-dominated Sorting Bee Colony Optimization Algorithm

Pratyusha Rakshit[1], Arup Kumar Sadhu[1], Preetha Bhattacharjee[2], Amit Konar[1], and Ramadoss Janarthanan[3]

[1] ETCE Dept.
[2] CSE Dept., Jadavpur University, Kolkata-700032, India
[3] Jaya Engineering College, Chennai
`{pratyushar1,arup.kaajal,pretb91}@gmail.com,`
`konaramit@yahoo.co.in, srmjana_73@yahoo.com`

Abstract. The paper provides a new approach to multi-robot box pushing using a proposed Non-dominated Sorting Bee Colony (NSBC) optimization algorithm. The proposed scheme determines time-, energy- and friction-optimal solution to the box-pushing problem. The performance of the developed NSBC algorithm is compared to NSGA-II in connection with the box-pushing problem and the experimental results reveal that the NSBC outperforms NSGA-II in all the experiments.

Keywords: multi-robot box pushing, friction compensation, non-dominated sorting bee colony optimization.

1 Introduction

Optimization problems with multi-objective functions are essential to many engineering problems. The box pushing problem, dealt with in this paper, is related to the well known “Piano Mover’s Problem”: given an arbitrary rigid polyhedral environment, find a continuous collision free path taking the object from a source configuration to a desired destination configuration.

In the work proposed in this paper, two similar robots have to locally plan the trajectory of motion of the box in a complex terrain where the robot’s workspace has different friction coefficients in different regions. In [2], it has been attempted to satisfy multiple objectives concerning minimization of both time and energy required in local trajectory planning of the robots by NSGA-II.

The work proposed in this paper is different in two main aspects. First of all, we have additionally considered the frictional forces offered by the contact surface to the box and it is also focused that the forces applied by robots must be sufficient to counteract to this frictional force. Secondly, the formulation of box- pushing as a multi-objective optimization problem, concerning minimization of time and energy required and the frictional force overcome by the robots while moving the box has been solved by a new algorithm, Non-dominated sorting bee colony optimization (NSBC), a sorting algorithm based on ABC, formulated by Karaboga et. al.

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 601–609, 2011.
© Springer-Verlag Berlin Heidelberg 2011

NSBC seems particularly suitable for this problem because of the following points: 1) better solution quality 2) not requiring big number of colony size to solve optimization problem with high dimensions because of using exploitive process efficiently to converge minima and explorative process to provide sufficient diversity in the population 3) increased search efficiency as not being sensitive to the search ranges. It has also been verified in the paper that NSBC has outperformed NSGA-II in trajectory planning of the robots in all the different types of workspace.

The remaining paper has been organized into sections as follows: In Section-2, a formulation of the problem is presented. In Section-3, a pseudo-code for solving the optimization problem using NSBC is provided and in Section-4, computer simulation for the problem has been laid down for comparison of results using NSBC and NSGA-II.

2 Formulation of the Problem

Suppose two robots R_1 and R_2 are applying forces perpendicularly at points E(x_e, y_e) and F(x_f, y_f) on the front edge BC of the box (Fig. 1). Let O (x_c, y_c) be the centre of gravity of the box . After being rotated by an angle α around the point $I(x_I, y_I)$ due to the forces applied by robots $R_1(F_{1r})$ and $R_2(F_{2r})$ the corresponding new co-ordinates of O, E and F become

$$\left.\begin{aligned} x_{inew} &= x_I(1-\cos\alpha) + x_i\cos\alpha - \sin\alpha\,(y_i - y_I) \\ y_{inew} &= y_I(1-\cos\alpha) + y_i\cos\alpha - \sin\alpha\,(x_i - x_I) \end{aligned}\right\} \text{ where,} i \in P = \{o, e, f\}. \quad (1)$$

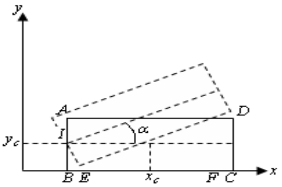


Fig. 1. Position of the box before (solid line) and after rotation (dashed line)

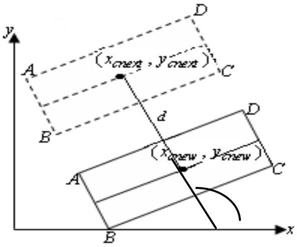


Fig. 2. Current (solid line) and next position (dashed line) of the box after rotation

Now, the box is displaced by d unit (Fig.2) due to transportation forces applied by $R_1(\ F_{1t})$ and $R_2(\ F_{2t})$. The new position of the centre of the gravity is given as

$$x_c' = x_{cnew} + d\cos\alpha\,;\ y_c' = y_{cnew} + d\sin\alpha \quad (2)$$

We, now form two objective functions concerning minimization of time and energy [2], which have three components each, time and energy required for rotation (t_1, E_1), for translation of the box to the next position (t_2, E_2), and predicted cost required for the transportation of the box from the next to the goal position through a distance s (t_3, E_3). The total time required (f_1) and total energy consumption $(f_2)$are obtained as

$$f_1 = t_1 + t_2 + t_3 \quad (3)$$

$$f_2 = E_1 + E_2 + E_3 \tag{4}$$

$$t_1 = \sqrt{\frac{2\alpha J}{T}},\ t_2 = \sqrt{\frac{2md}{F_{1t}+F_{2t}}},\ t_3 = k_t\sqrt{s};\ \ E_1 = T\alpha,\ E_2 = (F_{1t}+F_{2t})d\ ,\ E_3 = k_e s \tag{5}$$

where, J=mass moment of inertia, T= Torque, m=mass of the box, k_t and $k_e$are constants.

In order to keep the distance of the nearest obstacle dis_obs in the direction of movement as high as possible we introduce one penalty term in f_1.

$$f_1 = t_1 + t_2 + t_3 + f_{st}/dis\_obs \quad \text{with} \quad f_{st} = 500. \tag{6}$$

The third objective function is about minimization of frictional energy consumption during translation of the box in the workspace having different frictional coefficients in different regions (Fig. 3) and is denoted by, f_3 where

$$f_3 = F_c \times d \ \text{ where, } F_c = \text{Coulomb force of friction} = \textstyle\sum_{i=1}^{n} \mu_i \times m_i \times g. \tag{7}$$

where, n = total number of point contacts between the two surfaces in contact which is dependent on dimension of the box and μ_i and m_i represent the frictional coefficient of the surface and mass of the box at that point of contact.

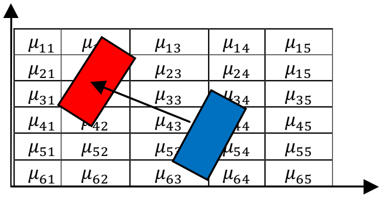


Fig. 3. Initial(blue) and final(red) position of the box in workspace with different frictional co-efficient

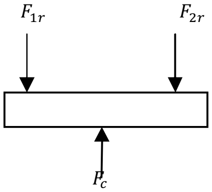


Fig. 4. Forces applied on the box by robots (F_{r1} and F_{r2}) and contact surface (F_c)

Considering the constraint that when the robots are rotating and translating the box, the combined force of the two robots must be greater than the frictional force offered by the surface in contact with the box, which is assumed to be acting on the centre of gravity of the box as (Fig. 4), we have the required condition as

$$F_{1r} + F_{2r} > F_c \tag{8}$$

So, with $d$unit displacement of the box equation (9) has been modified to

$$f_3 = \textstyle\sum_{i=1}^{n} \mu_i \times m_i \times g \times d + f_{fp}/(F_{1r} + F_{2r} - F_c) \tag{9}$$

where, g= acceleration due to gravity, and f_{fp} is a constant which is taken as 100. f_{st} and f_{st} are determined in a manner so that the terms on the right hand side of (9) 1nd (6) should be of same order of magnitude.

3 Solving Constraint Optimization Problem Using Non-dominated Sorting Bee Colony Optimization (NSBC)

In NSBC algorithm, the colony of artificial bees contains three groups of bees:

- Onlooker: waiting on a dance area for making decision to choose a food source.
- Employed: going to the food source visited by it previously.
- Scout: carrying out random search.

NSBC generates randomly distributed initial population P (g=0) of N_p solutions (food source positions) Each solution X_i (i=0, 1, 2,…, N_p-1) is a D dimensional vector. An employed bee produces a modification on the position (solution) in her memory depending on the local information (visual information) as stated by equation (2) and tests the nectar amount (fitness value) of the new source (new solution). After modification of food sources, all the parent and the child food sources are combined to form the merged population R_t i.e. $R_t = P_t \cup Q_t$, which is of size $2Np$. Here, the next population of food sources to be used by onlooker bees P_{t+1} is constructed by choosing the best Np solutions from the merged population R_t. Each solution is evaluated by using its rank as a primary criteria and crowding distance as secondary. The ranking is done based on the non-domination [2]. An onlooker bee evaluates the nectar information from all employed bees and chooses a food source depending on the probability value associated with that food source, p_i, calculated by the following expression:

$$p_i=dom_i/N_p \tag{10}$$

Where dom_i is the number of solutions dominated by the solution i. Now, onlooker bee produces a modification on the position (solution) in her memory and checks the nectar amount of the candidate source (solution). Neighborhood food source X_i' in the neighborhood of food source X_i is given as $X_i' = (x_{i0}, x_{i1}, x_{i2}, \dots, x_{i(j-1)}. x_{ij}'. x_{i(j+1)}, \dots, x_{i(D-1)})$. The value of x_{ij}'parameter in X_i' solution is computed using the following expression:

$$x_{ij}' = x_{ij} + u(x_{ij} - x_{kj})\ j \in [0, D-1], k \in [0, Np-1], k \neq i, u \in [-1,1] \tag{11}$$

The entire parent and the child food sources after getting modified by onlooker bees are combined to form the merged population R_t i.e.$R_t = P_t \cup Q_t$, which is of size $2Np$. As in case of employed bees, here also, using the methodology of non-dominated sorting the non-dominated food sources are found out from merged population of size $2Np$ to form the resulting population $P_{t+1}$of size Np. If a position cannot be improved further through a predefined number of cycles called ‘limit’, the food source is abandoned and is replaced by the scout randomly.

```
Pseudo code:
Input: Initial $(x_c, y_c)$ and final CG $(x_{cg}, y_{cg})$ of the box, points of
application of the two forces on the box by the two robots
$(x_e, y_e)$ and $(x_f, y_f)$ and a threshold value $\in$.
Output: Forces applied by the two robots to move the box and the
rotational angle $\alpha$.
```

```
Begin
Set: $x_{curr} \leftarrow x_c; y_{curr} \leftarrow y_c$;
Repeat until $||curr - G|| <\in$ // $curr$ = ($x_{curr}, y_{curr}$), G = ($x_{cg}, y_{cg}$)//
Call NSBC$(x_{curr}, y_{curr}, x_e, y_e, x_f, y_f; \alpha, x_I, y_I, F_{1r}, F_{2r})$;Move-to ($x_{curr}, y_{curr}$);
End.
Procedure NSBC ($x_{curr}, y_{curr}, x_e, y_e, x_f, y_f; \alpha, x_I, y_I, F_{1r}, F_{2r}$)
Begin
Initialize all food sources and place them in $P_t$; Set $R_t$=$\phi$;
Evaluate the fitness ($fit_i$) of the population.
For Iter=1 to Maxiter do
 For each employed bee do
  Produce a new solution $X_i^{'}$ from equation (11);
  Calculate its fitness value and add both $X_i$ and $X_i^{'}$ in $R_t$;
 End for;
 Call NS ($P_t, R_t$);
 For each onlooker bee do
  Select the food source $X_i$ depending on $p_i$ as in (10);
  Produce new solution $X_i^{'}$ using the same equation (11);
  Calculate its fitness value and add both $X_i$ and $X_i^{'}$ in $R_t$;
 End for;
 Call NS ($P_t, R_t$);Initialize abandoned food source by scout;
 Memorize the best solution obtained so far;
 End for;
 Update:
   1. ( $x_{cnew}, y_{cnew}$), ($x_{enew}, y_{enew}$), ($x_{fnew}, y_{fnew}$) using equation 1;
   2. $x_{curr} \leftarrow x_{curr} + dcos\theta$; $y_{curr} \leftarrow y_{curr} + dsin\theta$ ;
   3. $x_e \leftarrow x_e + dcos\theta$; $y_e \leftarrow y_e + dsin\theta$ ;
   4. $x_f \leftarrow x_f + dcos\theta$; $y_f \leftarrow y_f + dsin\theta$ ;
 Return.
Procedure NS ($P_t, R_t$)
Begin
Construct non-dominated front sets($F_1, F_2$--);Set:$P_{t+1}$= $\phi$ ,i=0;
Repeat until $|P_{t+1}| + |F_i| \le N_p$
   $P_{t+1}$= $P_{t+1} \cup F_i$; i=i+1;
Sort $F_i$ in descending order of crowding distance;
$P_{t+1}$= $P_{t+1} \cup$First ($N_p - |P_{t+1}|$) elements of $F_i$;
End.
```

4 Experiment and Computer Simulation

The experiment was carried out on a simulated environment on Intel Core 2 Duo processor architecture with clock speed of 2GHz. The structure of food source in NSBC used here is given below.

$F1r$	X_I	α	$F1t$	d	$\mu.m$

Since all food sources in the Pareto front are equally good, to select the one among many possible solutions, we normalize both time, energy and frictional loss for the individual food source in the Pareto front. Let $f_1(i)$, $f_2(i)$ and $f_3(i)$ be the measure of time, energy and frictional energy loss for the i-th food source in the Pareto front.

$$f_1^f(i) = \frac{f_1(i)}{\sum_{i=1}^{n} f_1(i)};\ f_2^f(i) = \frac{f_2(i)}{\sum_{i=1}^{n} f_2(i)};\ f_3^f(i) = \frac{f_3(i)}{\sum_{i=1}^{n} f_3(i)} \tag{12}$$

The above process is repeated for all food sources in the Pareto front and in each step of the movement the following product is taken.

$$P_i = (f_1^f(i) \times f_2^f(i) \times f_3^f(i)) \tag{13}$$

Now, the effective food source having smallest P_i is identified in each step to move the box by the robots. The significance of this work is that here we have considered frictional loss occurring during the push operation by the robots with non uniform distribution of frictional coefficient in the robots work space. In this work, the position of the obstacles is made to be user defined. The frictional coefficients can also be changed during the course of the program. The trajectory of the box translated by robots on a frictionless surface using NSBC is represented in Fig. 5.

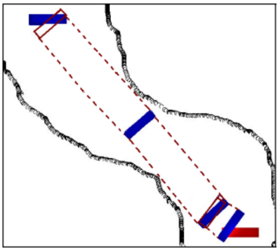


Fig. 5. Final configuration of the world map without friction compensation

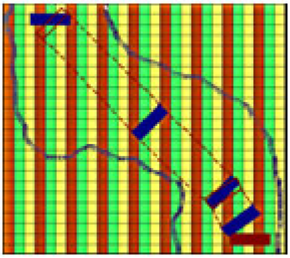


Fig. 6. Final configuration of the world map with static friction coefficients

Now, we have considered that different regions of the surface have got different static friction coefficients. Under such circumstances, the trajectory of the motion the box has been depicted in Fig. 6. In Fig. 7 we have shown the trajectory of the box when frictional coefficients of different region in the workspace have been changed by user during the runtime of the program. All the required information to govern the motion of the box at each step is given in following TABLE 1-4.

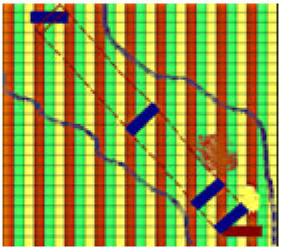


Fig. 7. Final configuration of the world map with dynamic friction coefficients

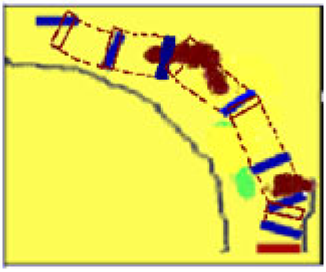


Fig. 8. Final configuration of the world map with friction compensation using NSBC requiring 5 steps

Table 1.

Position of Food Sources						
Steps	F_{1r}	X_I	α	F_{1t}	d	$\mu.\ m$
1	0.10	98.59	-0.753	0.10	400.00	2.316
2	11.93	253.35	0.024	14.76	290.29	0.713
3	5.55	412.23	-0.00043	0.10	99.53	2.736

Table 2.

Position of Centre of Gravity and Point of Rotation				
Steps	x_c	y_c	x_i	y_i
1	274.50	230.60	98.59	30.00
2	403.78	375.75	253.35	250.42
3	447.93	425.29	412.22	368.22

Table 3.

Time and Energy required and Frictional Force overcome by Robots in each Step			
Steps	*Time*	*Energy*	*Frictional Force*
1	1167.11	57.0077	23.167296
2	81.488	5721.355	0.699257
3	576.039	13.373	26.840160

Table 4.

Total Time and Energy required and Frictional Force overcome by Robots		
Total Time	*Total Energy*	*Total Frictional Energy*
1824.646	5791.7372	12141.501

In Fig. 8, it has been shown how friction compensation has been achieved by NSBC-based simulation. Initially the friction coefficient of the workspace is considered to be high ($\mu = 40$) as shown by yellow. But during the course of the program whenever less friction path has been provided ($\mu = 10$ in case of green and $\mu = 0.6$ in case of brown), robots always follow the less friction path.

The relative performance of NSBC- and NSGA-II-based simulation can be studied through the observation of the local trajectory planning by the robots and total time and energy required as well as total frictional energy overcome by robots to move the box from initial position to goal position and also by observing the number of steps required to reach their goals with NSBC and NSGA-II based algorithm as shown in following figures. NSBC seems to have marginally outperformed NSGA-II considering all the cases.

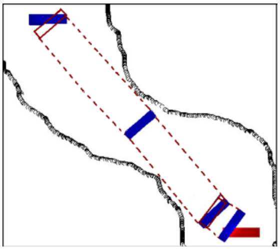


Fig. 9. Final configuration of the world map using NSBC without friction requiring 3 steps

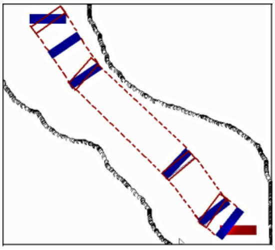


Fig. 10. Final configuration of the world map using NSBC without friction requiring 5 steps

It is apparent from Table V that both energy consumption and frictional loss have been reduced by a margin of 26.46%, 60.81%, 79.45% and 0%, 21.28%, 34.68% respectively due to NSBC. But time required has been increased by 33.03%, 41.93% and 31.94% due to complexity involved in NSBC.

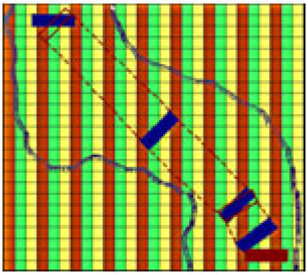


Fig. 11. Final configuration of the world map using NSBC with static friction requiring 3 steps

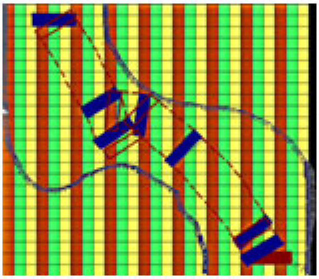


Fig. 12. Final configuration of the world map using NSGA-II with static friction requiring 6 steps

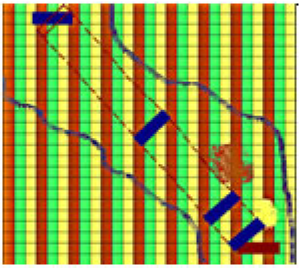


Fig. 13. Final configuration of the world map using NSBC with dynamic friction requiring 3 steps

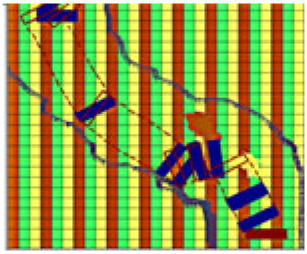


Fig. 14. Final configuration of the world map using NSGA-II with static friction requiring 8 steps

Table 5.

	Total Time and Energy required and Frictional Force overcome by Robots			
Case		*Total Time*	*Total Energy*	*Total Frictional Energy*
No friction	NSBC	248.318189	14617.80759	0
	NSGA-II	186.668398	19878.11030	0
Static friction	NSBC	522.226597	7091.913497	11876.664166
	NSGA-II	367.938531	18094.81926	15086.676629
Dynamic friction	NSBC	1824.646972	5791.737231	12141.501548
	NSGA-II	1382.992704	28193.83169	18587.646525

5 Conclusion

This paper provides a novel approach to handling box-shifting as a multi-objective optimization algorithm, and offers Pareto-optimal solutions in real time by utilizing the power of optimization of the NSBC program. The approach to the problem is unique and is different from the classical behavior based and perceptual cues based multi-robot box-shifting problems.

References

1. Karaboga, D.: An idea based on honey bee swarm for numerical optimisation, Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department (2005)
2. Chakraborty, J., Konar, A., Nagar, A., Das, S.: Rotation and translation selective Pareto optimal solution to the box-pushing problem by the mobile robots using NSGA-II. In: Evolutionary Computation 2009, IEEE Congress (2009)
3. Parker, L.E.: Adaptive heterogeneous multi-robot teams. Neurocomputing 28(1-3), 75–92 (1999)
4. Miyata, N., Ota, J., Arai, T., Asama, H.: Cooperative transport by multiple mobile robots in unknown static environments associated with real-time task assignment. IEEE Trans. Robot. Automat. 18(5), 769–780 (2002)
5. Sugar, T.G., Kumar, V.: Control of cooperating mobile manipulators. IEEE Trans. Robot. Automat. 18(1), 94–103 (2002)
6. Groß, R., Dorigo, M.: Evolving a Cooperative Transport Behavior for Two Simple Robots. In: Liardet, P., Collet, P., Fonlupt, C., Lutton, E., Schoenauer, M. (eds.) EA 2003. LNCS, vol. 2936, pp. 305–316. Springer, Heidelberg (2004)

Emotion Recognition from the Lip-Contour of a Subject Using Artificial Bee Colony Optimization Algorithm

Anisha Halder[1], Pratyusha Rakshit[1], Aruna Chakraborty[2],
Amit Konar[1], and Ramadoss Janarthanan[3]

[1] ETCE Department, Jadavpur University, Kolkata-32, India
halder.anisha@gmail.com, pratyushar1@gmail.com, konaramit@yahoo.co.in
[2] Department of Computer Science and Engineering,
St. Thomas College of Engineering and Technology, Kolkata, India
aruna_stcet@rediffmail.com
[3] Department of IT, Jaya Engg. College, Chennai
srmjana_73@yahoo.com

Abstract. This paper provides an alternative approach to emotion recognition from the outer lip-contour of the subjects. Subjects exhibit their emotions through their facial expressions, and the lip region is segmented from their facial images. A lip-contour model has been developed to represent the boundary of the lip, and the parameters of the model are adapted using artificial bee colony (ABC) optimization algorithm to match it with the boundary contour of the lip. An SVM classifier is then employed to classify the emotion of the subject from the parameter set of the subjects' lip-contour. The experiment was performed on 50 subjects, and the average case accuracy in emotion classification is found to be 86%.

Keywords: lip contour, artificial bee colony optimization algorithm, support vector machine.

1 Introduction

Emotion recognition is currently gaining importance for its increasing scope of applications in human-computer interactive systems. Several modalities of emotion recognition, including facial expression, voice, gesture and posture have been studied in the literature. Among the well-known methods of determining human emotions, schemes proposed by Ekman and Friesen, Kobayashi and Hara, Cohen et al. deserve special mention. However, we are afraid that there exists hardly any significant work [3] on emotion recognition by a single facial feature. This paper takes a serious attempts to recognize human emotion by considering the lip-contour of the subject.

In this paper we select a 6-segments lip-contour model, whose individual segments can be tuned to all typical non-overlapped lip-contours by controlling model parameters. An Artificial Bee Colony(ABC)optimization algorithm [1]

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 610–617, 2011.
© Springer-Verlag Berlin Heidelberg 2011

is used to match the model lip-contour with the segmented lip boundary of a subject. Experiments with 50 volunteers reveal that there exists a correlation between the lip-contour pattern of the individual, and a specific emotion experienced by the subject. This observation motivates us to design a classifier to map the extracted parameters of the lip-contour model on to the emotional space. We here, select a Support Vector Machine(SVM) classifier for classifying emotion.

The rest of the paper is divided into 5 sections. Section 2 offers the modeling issues of the human lip-contour.We introduce ABC algorithm in section 3. Methodologies are discussed in section 4. Experiments and results are given in section 5. The conclusions are listed in section 6.

2 The Proposed Lip-Contour Model

Until this time, there is no universally accepted model of lip-contour. In this paper, we start with the elementary kiss curve (Fig. 1), and modify it at different segments, as indicated in Fig. 2 to obtain an ideal model of the curve, capable of capturing most of the non-overlapped lip-contours in different emotional states. The basic equation of the kiss curve (Fig. 1) is given by

$$y^2 = (1 - x^2)^3, -1 \leq x \leq 1. \tag{1}$$

The curve returns both positive/negative values of y for each value of x. We, however, use the entire positive half of the curve, and a portion of the negative half. The remaining negative half is replaced by a parabola for better matching with lip profiles of the subjects. When the domain $-1 \leq x \leq +1$ is replaced by $[-l, +l]$, expression (1) is written as

$$y = \left(1 - \left(\frac{x}{l}\right)^2\right)^{\frac{3}{2}} \tag{2}$$

To determine all except the segment GA in Fig. 2, we scaled the right hand side of (2) and added one or more extra terms, as needed, and determine the parameters of the new curve by setting suitable boundary conditions corresponding to the corner points and axis crossings as listed in Table-1 and Table-2. The resulting parameters are also given in the tables.

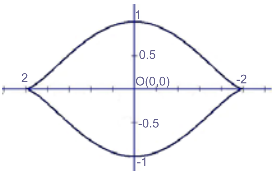


Fig. 1. The standard kiss curve

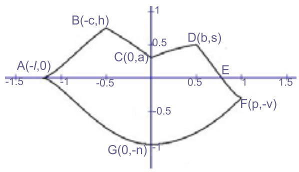


Fig. 2. The proposed model of the lip outer profile

Table 1. The Proposed Lip Segments with boundary conditions

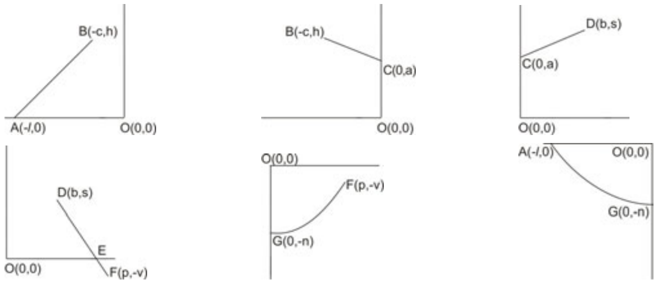


Table 2. Parametric Equations for the Proposed Lip Segments

Curve Segment	Presumed Equation	Boundary Condition	Parameter Obtained by Setting the boundary ions
AB	$y = a_1\left(1-\left(\frac{x}{l}\right)^2\right)^{\frac{3}{2}} + a_2$	$-l \leq x \leq -c$ $0 \leq y \leq h$	$a_1 = \frac{h}{\left(1-\left(\frac{c}{l}\right)^2\right)^{\frac{3}{2}}}$ $a_2 = 0$
BC	$y = a_3\left(1-\left(\frac{x}{l}\right)^2\right)^{\frac{3}{2}} + a_4 x$	$-c \leq x \leq 0$ $h \leq y \leq a$	$a_3 = a$ $a_4 = \frac{a\left(1-\left(\frac{c}{l}\right)^2\right)^{\frac{3}{2}}-h}{c}$
CD	$y = a_5\left(1-\left(\frac{x}{p}\right)^2\right)^{\frac{3}{2}} + a_6 x$	$0 \leq x \leq b$ $a \leq y \leq x$	$a_5 = a$ $a_6 = \frac{s-a\left(1-\left(\frac{b}{p}\right)^2\right)^{\frac{3}{2}}}{b}$
DEF	$y = a_7\left(1-\left(\frac{x}{p}\right)^2\right)^{\frac{3}{2}} + a_8 2$	$b \leq x \leq p$ $s \leq y \leq -v$	$a_7 = \frac{s+v}{\left(1-\left(\frac{b}{p}\right)^2\right)}$ $a_7 = 0$ $a_8 = -v$
FG	$y = a_9 x^2 + a_{10} x + a_{11}$	$p \leq x \leq 0$ $-v \leq y \leq -n$	$a_9 = \frac{n-v}{p^2}$ $a_{10} = 0$ $a_{11} = -n$
GA	$y = \pm\left(1-x^2\right)^{\frac{3}{2}}$	$0 \leq x \leq -l$ $-n \leq y \leq 0$	$y = -n\left(1-\left(\frac{x}{l}\right)^2\right)^{\frac{3}{2}}$

3 The ABC Algorithm

The artificial bee colony optimization (ABC) technique is a population based algorithm for numerical function optimization that draws inspiration from the stochastic behavior of foraging in bees.In ABC trial solution, the population directly affects the mutation operation since it is based on the difference of two members of the population. Hence by this method the information of a good member of the population is distributed among others due to the mutation operation and greedy selection mechanism employed to obtain a new member of the population. In the ABC, while the intensification process is controlled by the stochastic and the greedy selection schemes, the diversification is controlled by the random selection. In ABC algorithm, the colony of artificial bees contains three groups of bees:

- A bee waiting on a dance area for making decision to choose a food source is called an onlooker.
- A bee going to the food source visited by itself previously is named as employed bee.
- A bee carrying out random search is called a scout.

In ABC algorithm, the position of a food source represents a possible solution of the optimization problem and the nectar amount of a food source corresponds to the fitness of the associated solution. The number of employed bees and onlooker bees is equal to the number of solutions in the population. The ABC algorithm consists of following steps:

1. Generate randomly distributed initial population P (g=0) of N_p food sources where, each food source is of dimension D.
2. FOR each i-th employed bee do Generate neighborhood food source

   $$X_i^{/} = (x_{i0}, x_{i1}, x_{i2}, ..., x_{i(j-1)}, x_{ij}^{/}, x_{i(j+1)}, ..., x_{i(D-1)}) \tag{3}$$

   with

   $$x_{ij}^{/} = x_{ij} + u(x_{kj} - x_{ij}) \tag{4}$$

   where, u is a random variable in [-1,1] and $k \in (0, Np-1), k \neq i$
   and $j \in (0, D-1)$
   IF $fit(X_i^{/}) > fit(X_i)$ THEN $X_i = X_i^{/}$
   END FOR.
3. FOR each onlooker bee do Select food source X_i depending on probability

   $$p_i = fit(X_i) / \sum_{j=0}^{N_{p-1}} fit(X_j) \tag{5}$$

   Generate $X_i^{/}$ as in equation (3) and (4).
   IF $fit(X_i^{/}) > fit(X_i)$ THEN $X_i = X_i^{/}$
   END FOR.

4. If any food source is abandoned by employed bees, re-initialize the food source by scout bee.
   Repeat steps (1), (2), (3) and (4) until required condition is reached.

4 Methodology

4.1 Segmentation of the Lip-Contour

Several algorithms for the lip segmentation are available in the literature [2], [3]. In this paper, we, however, employ fuzzy k-means clustering algorithm to segment the lip region from the rest of the facial expression. In order to segment the lip region, we first localize the mouth region as shown in Fig. 3.a. Then a conversion from r-g-b to l-a-b color space is undertaken. The k-means clustering algorithm is applied next on this image to get three clusters. The three clusters are: skin, lip and teeth regions. The cluster with the highest intensity variance in l-a-b color space is declared as the lip region. Thus we select the lip cluster (Fig. 3.b) to determine the lip-contour.

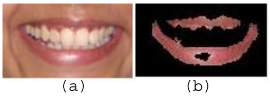


Fig. 3. The segmented mouth region obtained by FCM algorithm

4.2 Parameter Extraction of a Given Lip-Contour Using Artificial Bee Colony Optimization Algorithm

Artificial Bee Colony optimization(ABC) algorithm proposed by Karaboga and Basturk [1] offers promising solution to a global optimization problem. In this paper, we employ artificial bee colony as the optimization algorithm to determine the lip parameters of a given subject carrying a definite emotion.Given a finite set of selected points on the lip boundary of a segmented mouth region, and a model lip curve, we need to match the response of the model curve with the selected data points by varying the parameters of the model curve. The set of parameters for which the best matching takes place between the model generated data points and the selected lip boundary points are the results of a system identification procedure adopted here.

Let, $y = f(x)$ be the model curve. Then for all (x, y) lying on the curve, we obtain $G(x, y) = 1$, and for all points $y \neq f(x), G(x, y) = 0$. Let, $L(x, y) = 1$ for all valid data points on the outer boundary of a segmented lip. We use a performance evaluation metric J, where

$$J = \sum_{\forall x} \sum_{\forall y, y=f(x)} |G(x, y) - L(x, y)| \tag{6}$$

In ABC algorithm, we used J as the fitness functions, where we wanted to minimize J for all valid (x, y) on the lip boundary. The ABC considers 9-parameter food sources, finds out the neighborhood food source by mutation operation, and selects the best of the searched food source and the original food source to determine the parameter vector in the next iteration. This is done in parallel for NP number of parameter vectors, where NP is the population size. The algorithm is terminated when the error limit, defined by the difference of J's between the best of the previous and the current iteration is below a prescribed threshold. The best fit parameter vector is the parameter set of the best model lip-contour matched with a given lip boundary data points.

4.3 Emotion Classification from Measured Parameters of the Lip-Contour Model

It is noted from a large number of lip-contour instances that there exist at least two parameters of the lip model clearly distinctive of individual emotions. So, any typical machine learning/statistical classifier can be employed to classify the different emotional status from the parameter of lip-contour. In this paper, we use Linear Support Vector Machine (SVM) [2] classifier for emotion classification from the lip data.

In our basic scheme, we employed five SVM networks, one each for joy, anger, sadness, fear and relaxation. The i^{th} SVM network is trained with all of the training instances of the i^{th} class with positive levels, and all other training instances with negative levels. The decision logic box driven by the five SVM networks ultimately recognizes the emotion corresponding to the supplied feature vector X.

The decision logic works in the following manner. If only one input of the decision logic is +1, it infers the corresponding class number at the output. For, example, if the SVM-disgust only generates a +1, the output of the decision logic will be the class number for the emotion class: disgust.

When more than one input of the decision logic is +1, the decision is taken in two steps. First, we count the number of positive instances falling in the small neighborhood of the given pattern for each emotion class with its corresponding SVM-output +1. Next, the emotion class with the highest count is declared as the winner. The decision logic thus takes decision based on the principle of "majority voting", which is realized here by the count of positive instances in the neighborhood of the given test pattern. The "neighborhood" here is defined as a 9-dimensional sphere function around a given test pattern, considering it as the center of the sphere. The radius of the sphere is determined from the measurements of standard deviation in the individual features. The largest among the standard deviations for all the features is considered as the radius of the "neighborhood" in the data points, representing positive instances in a given emotion class. The radius of different emotion classes here thus is different. This, however, makes sense as the data density (per unit volume in 9-dimensional hyper space) for different emotion classes is non-uniform.

5 Experiments and Results

The experiment has two phases. In the first phase, we determine weight vectors of 5 SVM classifiers, each one for one emotion class, including anger, disgust, fear, happiness and sadness. We had 50 subjects, and for each subject we obtained 10 facial expressions for 10 different instances of each emotion. Thus for 5 emotions, we had 50 facial expressions for individual subjects. Three out of 10 instances of emotional expression are given in Table 3 for a subject.

Now, for each facial expression given in Table 3, we segmented the mouth region by Fuzzy k-means clustering algorithm, and determined the optimal lip-parameters: b, c, l, p, v, n, a, h and s by adapting the model lip-contour with

Table 3. Facial Expression for Subject 1 for Different Emotions

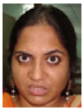
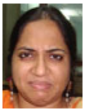
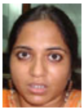
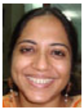
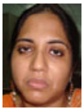
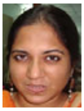
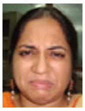
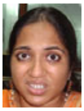
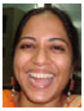
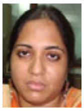
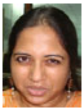
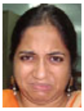
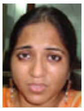
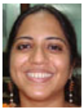
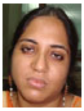
	Emotion				
Instances	Anger	Disgust	Fear	Happiness	Sadness
1.					
2.					
3.					

Table 4. Lip Parameters for Subject 1 for Different Emotions

Emotion	Instance	b	c	l	p	v	n	a	h	s
HAPPY	1	55	63	226	228	55	194	25	14	0
	2	46	80	221	249	36	241	44	52	44
	3	45	69	221	228	49	156	49	54	49
SAD	1	39	38	170	149	38	64	64	81	75
	2	37	45	171	152	38	45	45	66	53
	3	38	40	164	159	17	67	67	79	74
FEAR	1	48	49	148	142	18	116	46	60	52
	2	45	54	151	167	26	141	38	61	54
	3	44	47	143	151	29	126	36	54	44
DISGUST	1	41	40	190	168	26	96	10	24	19
	2	35	36	187	159	22	92	12	27	23
	3	39	48	176	165	30	98	12	29	18
ANGER	1	36	48	147	143	33	133	31	49	40
	2	32	48	161	168	25	134	26	45	35
	3	33	39	140	144	25	127	42	53	49

the outer boundary of individual segmented lips to have an optimal matching between the two. This matching was performed by ABC algorithm. Table 4 shows the result of lip-parameters obtained from Table 3. The weight vectors for the SVM classifiers for individual emotion of a subject are then identified. This is done by first preparing a table with 10 positive and 40 negative instances for each emotion classes of a subject. The weight vector for the given SVM-classifier for the emotion class is determined in a manner, so that all the positive and negative instances are separated with a margin of $\frac{2}{\|W\|}$. This is done for all individual subjects separately.

The second phase of the experiment starts with an unknown facial expression of a known subject. We first obtain mouth region from the image by Fuzzy K-means clustering, and determine lip-parameter by ABC using the model-lip. Now, we feed the lip-parameters to all the 5 SVM classifiers for the person concerned. The output of one or more SVM-classifiers may be +1. The decision logic then determines the emotion class of the unknown pattern.

6 Conclusion

The chapter proposed a new approach to emotion classification from the lip-contour of the subjects experiencing a specific emotion. Experiments with large number of subjects confirm that the proposed model can capture most of the experimental lip-contour for a specific emotive experience of the subject. The ABC algorithm used here is very fast and robust and thus can easily determine the parameters of the lip-contour. The SVM classifier, which is already an established tool for pattern classification with high accuracy has been utilized here for classifying lip parameters onto emotions. Experiments here too confirm that the percentage accuracy in classification of emotion on an average is 86% as obtained from the data set of 50 Indian subjects each having ten frames per emotion.

References

1. Karaboga, D., Basturk Akay, B.: Artificial Bee Colony Algorithm on Training Artificial Neural Networks. In: IEEE 15th Signal Processing and Communications Applications, pp. 1–4 (June 2007)
2. Konar, A., Chakraborty, A.: Emotional Intelligence: A Cybernetic Approach. Springer, Heidelberg (2009)
3. Bouvier, C., Coulon, P.-Y., Maldague, X.: Unsupervised Lips Segmentation Based on ROI Optimisation and Parametric Model. U.Stendhal 46 av. F. Viallet, France and LVSN, University Laval, Canada in IEEE (2007)

Software Coverage : A Testing Approach through Ant Colony Optimization

Bhuvnesh Sharma, Isha Girdhar, Monika Taneja, Pooja Basia, Sangeetha Vadla,
and Praveen Ranjan Srivastava

Birla Institute of Technology and Science, Pilani Campus, India
{bhuvnesh.bits,ishagirdhar,monicks1,pooja.basia,san3003.v,
praveenrsrivastava}@gmail.com

Abstract. Software Testing is one of the most important parts of the software development lifecycle. Testing effectiveness can be achieved by the State Transition Testing (STT) and path testing which is commonly used for carrying out functional and structural testing of software systems. The tester is required to test all possible transitions and paths in the system under built. Aim of the current paper is to present an algorithm for generation of test sequences for state transitions of the system as well as path generation for CFG of the software code using the basic property and behavior of the ants. This novel approach tries to find out all the effective (or can say optimal) paths and test sequences by applying ant colony optimization (ACO) principle using some set of rules. This algorithm tries to give maximum software coverage with minimal redundancy.

Keywords: Software Testing, Ant Colony Optimization (ACO), Genetic Algorithm (GA), State Transition Testing (STT), Test Data, Test Sequence, Control Flow Graph (CFG).

1 Introduction

Software Engineering [1] is an engineering discipline which focuses on developing high-quality software systems which are cost effective. It is a profession where in designing, implementation, and modifications of software are involved. Software Development Life Cycle [1] is a process of developing a system which involves different steps like investigation, analysis, design, implementation and maintenance. Through software testing an organization can gain consumers' confidence towards the system [1]. Software testing is a labour intensive and very expensive task. It accounts almost 50 % of software development life cycle [2] [3]. Many testing tools and methodologies have been emerging in the field of software testing. Code coverage analysis is one such methodology, which helps in discovering the instructions in a software program that have been executed during a test run and helps in discovering how the testing can be further improved to cover more number of instructions during testing of a software program [2].

This paper proposes an algorithm with tool, named **ESCov** (Efficient Software COVerage), which uses an ACO technique [4] to generate the optimal paths, to ensure

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 618–625, 2011.
© Springer-Verlag Berlin Heidelberg 2011

maximum coverage of structural as well as behavioral testing with minimal redundancy. The work is divided into two phases. In the first phase, maximum code coverage will be achieved via functional testing. For this optimized test sequences are generated by applying the ESCov, on the directed tree graph generated from the state chart diagram of the system. After that for the obtained test sequences test data is generated manually while in the second phase, the same ESCov is applied on the CFG of the software code under test. Finally, test data is applied on the generated test sequences obtained in the second phase to validate the maximum coverage.

2 Background Work

Software testing is done with many approaches such as Particle Swarm Optimization, Genetic Algorithm, and Ant Colony Optimization etc. in which ACO is one of the effective approaches. A hybrid approach using ACO and a markov software usage model [5] can produce some better results for deriving a set of independent test paths for a software system. McMinn [6] discussed application of ACO for finding sequences of transitional statements in generating test data under evolutionary testing. Another approach for state based testing using an ACO is represented [7] in which only states are considered and not complete transitions. Many AI based techniques like Neural Networks, the black-box data processing structure is very complex and it provides a slow convergence speed [8]. Although the search in genetic algorithms is not exhaustive, the algorithm fails to find a global optimum of the fitness if get stuck in local extreme [9].

Ant colony optimization is inspired by foraging behaviors of ant colonies, and target discrete optimization problem [4]. Real ants coordinate with each other by dropping and sensing a chemical level known as pheromone on the paths. Selection of path by ants depends on stochastic or probability theory. The main idea is the self organizing principle, which allows the highly coordinated behavior of real ants can be applied with the artificial agents, which collaborate to solve the complex computational problems. In the next section this paper proposed solution regarding total software coverage.

3 Proposed Algorithm

Purpose of the proposed Algorithm is providing the optimal path coverage for state transition diagram as well as CFG of the software under test. Selection of path depends upon the probability of the path. Ant will choose the path where the probability value is high. The probability value of path depends upon: probability value calculated by ant at each node for which it has pheromone value (τ_{ij}), which helps other ants to make decision in the future, and heuristic information (η_{ij}) of the path, which indicates the visibility of a path for an ant at the current vertex. In some cases if there are equal probabilities the ant will select any feasible path randomly. The architecture of the proposal is depicted in Figure 1.

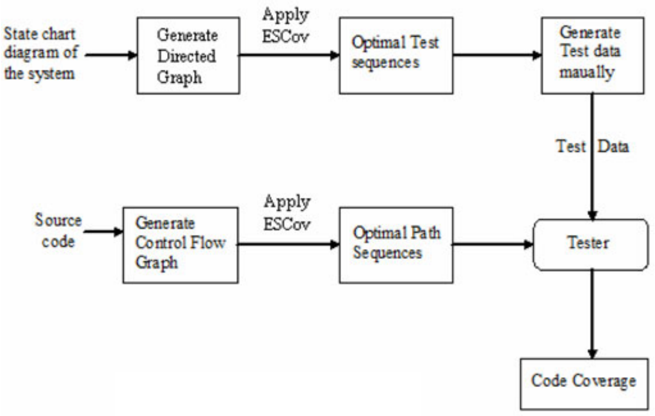


Fig. 1. Architecture of ESCov

In ESCov ant will calculate the probability based on the pheromone value and heuristic value. After the selection of a particular path ant update the pheromone level as well as heuristic value. Pheromone level is increased according to last pheromone level but heuristic value is decreased according to the last heuristic value.

An ant p at a vertex ‘i’ and another vertex ‘j’ which is directly connected to “i”, it means there is a path between the vertices ‘i’ and ‘j’ i.e. (i→j). The use of variables and sets used in the ESCov is given below:

1. Edge set: E = {(a, b), ()…} represents the set of all the edge(s) present in the graph.

2. Vertex set: V = {a, b…} represents the set of all the nodes present in the graph.

3. Track set: T = {{a, b…}, {a, c…}…} represents the set of traversed path(s) in the graph.

4. Pheromone value: $\tau = \tau_{ij}$ (p) represents the pheromone level on the edge (i, j) from current vertex ‘i’ to next vertex ‘j’ for ant ‘p’. The pheromone level is updated after the particular edge traversed. This pheromone helps other ants to make decision in future.

5. Heuristic value: $\boldsymbol{\eta} = \eta_{ij}$ (p) indicates the desirability of a path for an ant at current vertex ‘i’ to vertex ‘j’ for ant ‘p’.

6. Probability: Selection of path depends on probabilistic value of path because it is inspired by the ant behavior. Probability value of the path depends upon pheromone value τij (p) and heuristic information ηij (p) of path for ant p. There are α, β two more parameter which used to calculate the probability of a path. These parameters α and β control the desirability versus visibility. α and β are associated with pheromone and heuristic value of the paths respectively. In the ESCov, values taken for α and β is 1.

An algorithm is proposed for generating optimal paths for state transition diagram and CFG as well.

Algorithm for ant p

```
Step 1: Initialize all parameter
1.1 Set heuristic Value (η): for every edge initialize heuristic value η =1.
1.2 Set pheromone level (τ): for every edge initialize pheromone value τ =1.
1.3 α=1, β= 1
Step 2: Run the algorithm
2.1 WHILE indegree of end node(s) is not empty DO
2.2 Set current = start node
2.3 LOOP
        Evaluation at vertex i
2.3.1 Update the Track Set–
Push the current vertex i into the track set T
2.3.2 At the ith node Get next nodes
    (check if there is a 'self loop')
2.3.3 IF (there is a 'self loop') THEN
2.3.3.1 Select that edge regardless of the probability
2.3.3.2 Move to the selected node.
2.3.3.3 Enter it in track set T.
2.3.3.4 Delete the edge from E.
    (check if an edge is there to 'End node'.)
2.3.4 ELSE IF (there is a direct edge to 'end node') THEN
2.3.4.1 Select that edge regardless of the probability
2.3.4.2 Select the node
2.3.4.3 Enter it in track set T
2.3.4.4 IF outdegree > 1 AND
            (no of end nodes) >1 THEN
        Delete the edge from E
2.3.4.5 ELSE IF indegree > 1 THEN
        Delete the incoming edge from E
2.3.4.6 ELSE IF indegree = 1 AND
            outdegree = 1 THEN
        Delete the incoming and outgoing edges from E
2.3.4.7 END IF
        BREAK (LOOP)
2.3.5 ELSE
2.3.5.1 Calculate the probability of all possible edges
    $P_{ij} = \frac{[\tau_{ij}]^{\alpha} \times [\eta_{ij}]^{-\beta}}{\sum_{1}^{k} [[\tau_{ik}]^{\alpha} \times [\eta_{ik}]^{-\beta}]} \forall k \in feasibleset(F(p))$
2.3.5.2 Select the edge with max probability.
2.3.5.3 Select the node.
2.3.5.4 Update pheromone level and desirability factor of edge i,j as
New $\tau_{ij}$ = 2 * curr pheromone level of that edge
New $\eta_{ij}$ = curr heuristic value of that edge / 4
2.3.5.5 SET current = selected node
2.4 IF end node is not reached continue LOOP at Step 2.3
2.5 End While
```

4 Demonstration of Proposed Algorithm

The example work is divided into two phases: The basis of software requirement model, a state transition model constructed for a system, the algorithm is applied on directed graph under state transition diagram of the system in first phase and in second phase a CFG of the program code should be considered as input. In both the cases, an ant start from start node which would be specified by the user and it can generate a sequence of test cases. Test sequence depends upon the feasibility of path from the current vertex to other vertices and accordingly decision for further proceeding, and in the end, it give the optimal test sequences for the module under test. Here optimal means maximum coverage with minimal redundancy.

For the program taken under test, the state transition diagram and corressponding directed graph are given as below(for example) in Figure 2 and 3 respectively.

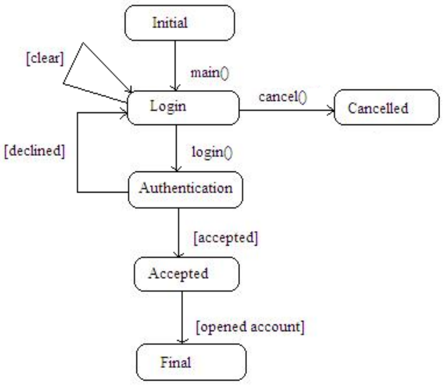


Fig. 2. The state transition diagram

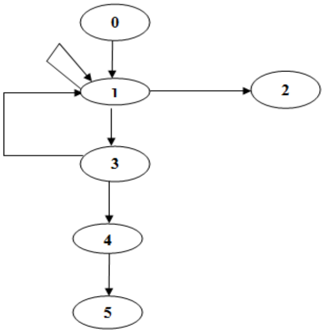


Fig. 3. The directed graph for the STD

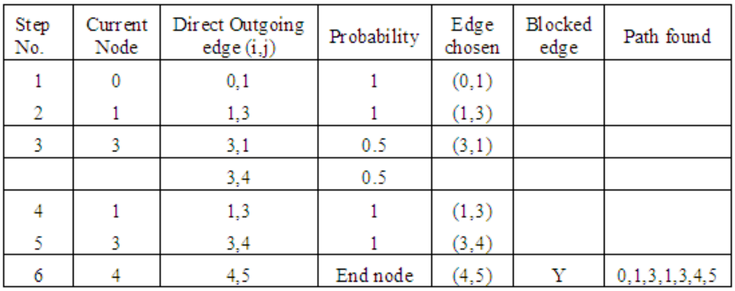
The ESCov is applied on given diagram and the test sequences obtained alongwith the values calculated at each node by the ants is given in tables (for respective ants).

At node 0 (starting node) there is only one outgoing edge to node 1. Ant move to the next node since there is no need of selection. At node 1 there are three outgoing edges (1, 1) self loop, (1, 2) and (1, 3). The ant calculates the probability value using the probability formula for each edge. All edges have same probability vaue of 0.33. So ant select any edge randomly say it chooses (1,1), update the pheromone level and heuristic value and move to the node 1. Now edge (1, 1) be blocked according to the algorithm. Now at node 1 ant is left with only two options (1, 2) and (1,3). Since 2 is the end node and the algorithm makes the ant to move to the end node if it is directly connected to the current node, ant chooses (1, 2). Since the end node is encountered, the first ant stops, and the path can be calculated by traversing the edges and nodes covered by the ant.

Ant 1 take the path mentioned in the table from the start node to the end node.

Step No.	Current Node	Direct Outgoing edge(i,j)	Probability	Edge chosen	Blocked edge	Path found
1	0	0,1	1	(0,1)	N	
2	1	1,1	Self loop	(1,1)	Y	
	1	1,2				
	1	1,3				
3	1	1,2	End-node	(1,2)	Y	0,1,1,2
	1	1,3				

Similarly, the second ant generate a path from the starting node to the end node mentioned in the table



Step No.	Current Node	Direct Outgoing edge (i,j)	Probability	Edge chosen	Blocked edge	Path found
1	0	0,1	1	(0,1)		
2	1	1,3	1	(1,3)		
3	3	3,1	0.5	(3,1)		
		3,4	0.5			
4	1	1,3	1	(1,3)		
5	3	3,4	1	(3,4)		
6	4	4,5	End node	(4,5)	Y	0,1,3,1,3,4,5

Test sequences generated by the two ants for the state transition diagram of the program under test are:

Path 1: 0→1→1→2 & *Path 2*: 0→1→3→1→3→4→5

Here the ants have generated two test sequences and covered all nodes as well as edges so the coverage is 100%.

Now the algorithm be applied to the CFG of the code and optimal paths would be generated. For the code under test (for example), the CFG is given in Figure 4. The path sequences obtained by applying the ESCov on the CFG are given in the tables given for individual ants.

The optimal paths generated by ESCov are

Path1:0→1→2→3→4
Path2:0→1→11→12→13
Path 3:0→1→2→3→5→6→3→5→6→7→8→10
Path4:0→1→11→12→14
Path5: 0→1→2→3→5→6→3→5→6→7→9→10

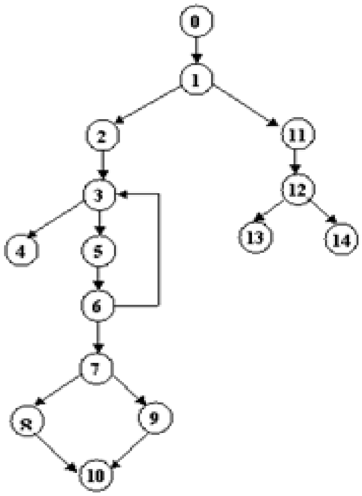
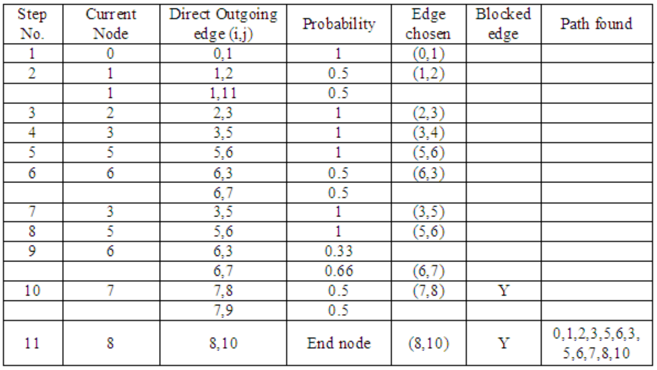


Fig. 4. CFG for source code

Illustration of path table for Ant 3 is shown in tabular representation as below (for example):



Step No.	Current Node	Direct Outgoing edge (i,j)	Probability	Edge chosen	Blocked edge	Path found
1	0	0,1	1	(0,1)		
2	1	1,2	0.5	(1,2)		
	1	1,11	0.5			
3	2	2,3	1	(2,3)		
4	3	3,5	1	(3,4)		
5	5	5,6	1	(5,6)		
6	6	6,3	0.5	(6,3)		
		6,7	0.5			
7	3	3,5	1	(3,5)		
8	5	5,6	1	(5,6)		
9	6	6,3	0.33			
		6,7	0.66	(6,7)		
10	7	7,8	0.5	(7,8)	Y	
		7,9	0.5			
11	8	8,10	End node	(8,10)	Y	0,1,2,3,5,6,3, 5,6,7,8,10

Here the ants have generated five paths and covered all nodes as well as edges so the coverage is 100% with minimal redundancy since no path is repeated.

5 Analysis

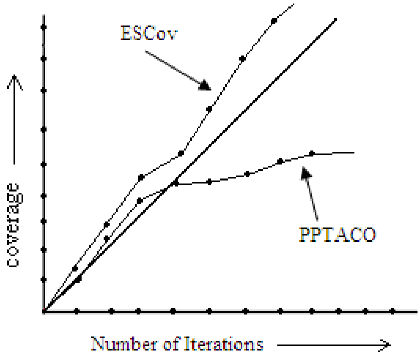
The main proposed advantage of the algorithm is the maximum coverage with minimal repetition. It is tried to be achieved by avoiding self loops to be executed repeatedly reducing the redundancy. The time complexity in best case is O (n*log n), but the same in average or worst case is O (n^2). There is a constraint in the ESCov in case of functional testing; it only captures single transition in between two states. If there are more than one transition in between two states, then only one is considered and rest are left untraversed.

6 Experimentation, Results and Discussion

The ESCov, to prove its essence, is compared with predefined tools (say Q) for functional coverage [11] and for structural coverage (PPTACO) [10].

Functional
This paper used a *class management system state machine diagram* [11] as case study on which it gave 4 possible paths for transitions given in table as reported in [11]. The ESCov is then applied on the state chart diagram of the same case study which now produces 3 independent paths covering all the states and hence gives total coverage. Therefore, it shows the ESCov provides minimal number of repetitions producing maximum coverage.

Structural
The PPTACO tool along with path generation also calculates strength [10] for prioritizing the paths to be covered. But as shown the path generation algorithm using ACO has repetition in it but ESCov provides the same coverage with minimum repetition.



The above graph shows the difference that the number of iterations increases if there is more scope of coverage. Every iteration gives a different path in the ESCov minimizing the redundancy.

7 Conclusion

This paper proposes a model and tool named 'ESCov' for test sequence generation for state based testing and optimal path generation for structural testing using ant colony optimization. The result that is produced by applying proposed method is very encouraging. To model the system state chart diagram and CFG are taken and the algorithm is applied over them. After successful execution of algorithm, it shows path sequence, which gives maximum coverage and minimum redundancy. This algorithm may be very useful and helpful for the testers in the software industry. A number of extensions and applications of this model may be possible by using the different meta-heuristic techniques.

References

[1] Sommerville, I.: Software Engineering, 8th edn. Pearson Edition (2009)

[2] Mathur, A.P.: Foundation of Software Testing, 1st edn. Pearson Education (2007)

[3] Myers, G.: The Art of Software Testing, 2nd edn., p. 234. John Wiley & Son. Inc. (2004)

[4] Dorigoa, M., Stutzle, T.: Ant colony optimization, The Knowledge Engineering Review, vol. 20, pp. 92–93. Cambridge University Press, New York (2005)

[5] Doerner, K., Gutjahr, W.J.: Extracting Test Sequences from a Markov Software Usage Model by ACO. In: Cantú-Paz, E., Foster, J.A., Deb, K., Davis, L., Roy, R., O'Reilly, U.-M., Beyer, H.-G., Kendall, G., Wilson, S.W., Harman, M., Wegener, J., Dasgupta, D., Potter, M.A., Schultz, A., Dowsland, K.A., Jonoska, N., Miller, J., Standish, R.K. (eds.) GECCO 2003. LNCS, vol. 2724, pp. 2465–2476. Springer, Heidelberg (2003)

[6] McMinn, P., Holcombe, M.: The State Problem for Evolutionary Testing. In: Cantú-Paz, E., Foster, J.A., Deb, K., Davis, L., Roy, R., O'Reilly, U.-M., Beyer, H.-G., Kendall, G., Wilson, S.W., Harman, M., Wegener, J., Dasgupta, D., Potter, M.A., Schultz, A., Dowsland, K.A., Jonoska, N., Miller, J., Standish, R.K. (eds.) GECCO 2003. LNCS, vol. 2724, pp. 2488–2500. Springer, Heidelberg (2003)

[7] Li, H., Lam, C.P.: An Ant Colony Optimization Approach to Test Sequence Generation for State based Software Testing. In: Proceedings of the Fifth International Conference on Quality Software (QSIC 2005), pp. 255–264 (2005)

[8] Briand, L.C.: On the many ways Software Engineering can benefit from Knowledge Engineering. In: Proc. 14th SEKE, Italy, pp. 3–6 (2002)

[9] Pedrycz, W., Peters, J.F.: Computational Intelligence in Software Engineering. World Scientific Publishers (1998)

[10] Srivastava, P.R., Baby, K.M.: An Approach of Optimal Path Generation using Ant Colony Optimization, pp. 1–6. IEEE-TENCON, Singapore (2009) ISBN-978-1-4244-4546-2

[11] Doungsa-ard, C., Dahal, K., Hossain, A., Suwannasart, T.: An Improved Automatic Test Data Generation from UML State Machine Diagram. In: ICSEA (2007)

Short Term Load Forecasting Using Fuzzy Inference and Ant Colony Optimization

Amit Jain[1], Pramod Kumar Singh[2], and Kumar Anurag Singh[2]

[1] IIIT Hyderabad, Hyderabad, India
[2] ABV-IIITM, Gwalior, India

Abstract. The short term load forecasting (STLF) is required for the generation scheduling and the economic load dispatch at any time. The short term load forecast calculates the power requirement pattern for the forecasting day using known, similar previous weather conditions. This paper describes a new approach for the calculation of the short term load forecast that uses fuzzy inference system which is further optimized using an Ant Colony Optimization (ACO) algorithm. It takes into account the load of the previous day, maximum temperature, average humidity and also the day type for the calculation of the load values for the next day. The Euclidean norm considering the weather variables and type of the day with weights is used to get the similar days. The effectiveness of the proposed approach is demonstrated on a typical load and weather data.

1 Introduction

Forecasting is the establishment of the future expectations by analysis of the past data, or formations of the opinions; it helps in planning and operation of the system. One of the major roles of the forecasting is in the better utilization of resources and the estimation of the benefits out of it. In general, forecasting methods may be broadly classified as (i) quantitative and (ii) qualitative. The quantitative techniques include simple regression, multiple regression, time trends, moving averages etc [1]. The qualitative methods include Delphi method, Nominal group technique, Jury of executive opinion, scenario projection etc.

The rapid growth of demand for energy in the last few decades and the depletion of the fossil fuel resources have given an impetus to the development of optimal energy planning in the power system utilization. In particular, the short term load forecasting (STLF) has gained importance and has put up greater research challenges due to the recent trend of the deregulation of the electricity industry. In case of the real time dispatch operation, any forecasting error causes more electricity purchasing cost or breaking penalty cost to keep up the electricity supply and the consumption balance. Several methods for the short term load forecasting are applied like the time series, regression analysis, exponential smoothing etc [2].

The load of the power systems is dependent on the weather conditions and the social conditions. The weather conditions include temperature, humidity, chill factor and wind speed etc., whereas the social conditions include, e.g., the holiday, working hours, demographic changes and changes in the living standards. Heinemann et al. [3]

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 626–636, 2011.
© Springer-Verlag Berlin Heidelberg 2011

have shown that the ratio of the summer to winter system peak load increased from 90% in the year 1953 to 111% in the year 1964. The daily peak load can be mainly divided into two types of load that are basic load and the weather sensitive load [4]. The system load has been responsive to changes in the weather conditions; hence the system operators must be prepared for summer afternoons having the sudden large increase in the demands. A large percentage of summer weather sensitive loads depend on the relative degree of comfort or discomfort of the human beings. The treatment of winter daily system peak loads would be very similar in details [5]. To make such forecasts, it is necessary to examine the history of each component to establish the trends.

The above factors complicate the load forecasting. We observe that, in the literature, multiple linear regression analysis based on the value of the previous load has traditionally been used for load forecasting. Recently, methods based on artificial intelligence, e.g., artificial neural networks, fuzzy logic, have also been discussed in the literature [6], [7], [8], [9], [10]. The artificial neural network does to produce promising result for STLF as it fails to respond properly with respect to rapid fluctuations in the data. Though there are many weather variables which affect the load, the major deciding variables are the temperature and the humidity [11]. In this paper, the proposed model takes into account the past month's temperature and humidity, load of the previous hour, temperature of the present hour, temperature of the previous hour, average humidity of the present as well as the previous hour and the similar day types.

This paper is organized as follows. Section 2 includes the data analysis part. The proposed forecasting method is presented in Section 3. Section 4 presents simulation results and is followed by the conclusion in Section 5.

2 Variables Affecting the Load Pattern

The analysis is carried out on data containing hourly values of the load, and average temperature and average humidity during that hour. A relationship between the load and the weather variables is presented here.

2.1 Variation of Load with Temperature

One of the most important weather variables that influence the load is temperature. A scatter plot of the given load and the hourly temperature values is presented in Fig. 1. It shows a positive correlation between the load and the temperature, i.e., demand of load increases as the temperature increases.

2.2 Variation of Load with Humidity

Humidity is another important weather variable which affects the load. Fig. 2 represents a scatter plot of the average load and the average humidity. Here too, it is observed that the correlation between the load and the humidity is positive, i.e., the demand of load increases as the humidity increases.

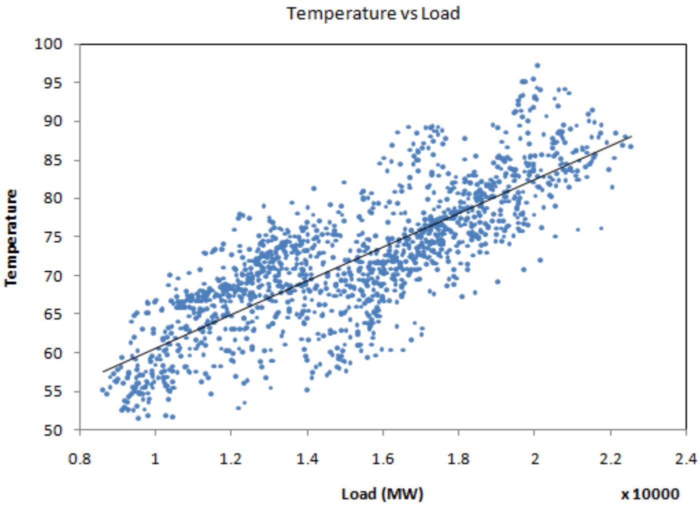


Fig. 1. Plot between average load Vs average temperature

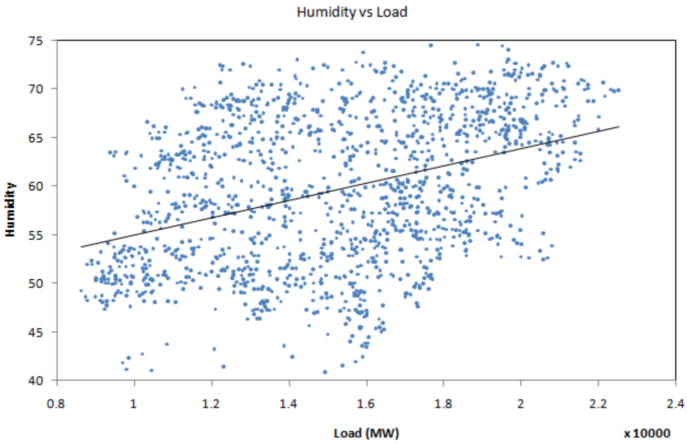


Fig. 2. Plot between the average load Vs average humidity

2.3 Relation between the Present and the Previous Hour Load

The present hour load is dependent on the previous hour load as well as the load of the same hour of the previous day. Hence, in multiple linear regression models the effect of the previous hour load has also been considered. It is assumed that there is no significant change in the seasonal weather in one day and also not significant change in temperature or humidity in one hour. Accordingly, the pattern of load consumption will also be similar to the previous load consumptions. Fig. 3 represents variation of the load consumption with the hours. It shows a cyclic behavior of the load consumption pattern; load consumption in weekdays is different from that of the weekend.

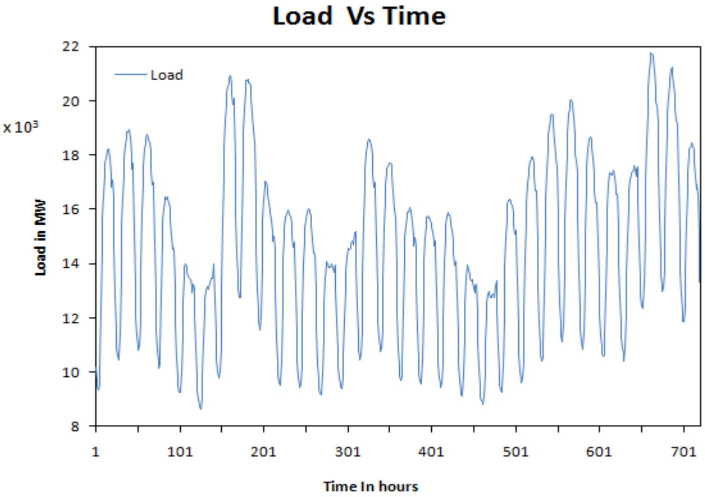


Fig. 3. The load consumption pattern Vs the hours

3 Short Term Load Forecasting Using Fuzzy Inference Followed by Ant Colony Optimization

3.1 Similar Day Selection

To calculate the similarity between the forecast day and the searched previous days, the Euclidean norm with the weight factors is used. The value of the Euclidean norm is inversely proportional to the similarity, i.e., lesser is the value of the Euclidean norm more similar is that particular day to the forecast day. The Euclidean norm uses the maximum temperature, average humidity and the day type, each of which is associated with the weights. It gives a better method of the selection of the similar days as it contains the temperature, the humidity and also the day type. The weight factors associated with the temperature, humidity and the day type are calculated based on the historic data by using regression analysis on it. The expression for the Euclidean norm is given by:

$$EN = (w_1 * (\Delta T)^2 + w_2 * (\Delta H)^2 + w_3 * (\Delta D)^2)^{0.5} \tag{1}$$

$$\Delta T_{max} = T_{max} - T_{max}^{p} \tag{2}$$

$$\Delta H_{avg} = H_{avg} - H_{avg}^{p} \tag{3}$$

$$\Delta D = D - D^{P} \tag{4}$$

Here, T_{max} and H_{avg} are maximum temperature and average humidity respectively for the forecast day, T_{max}^{p} and H_{avg}^{p} are maximum temperature and average humidity of the searched previous days. The weight factors w_1, w_2 and w_3 are evaluated using the regression analysis on the historical data. The weights are used in the formula of the Euclidean norm for the scaling purpose.

3.2 Fuzzy Inference System

The Euclidean norm alone should not be used for the forecasting process as it can have large value of mean absolute percentage error. Assuming the same trends of the relationship between the forecast day and the previous similar days as that of the current day forecast and its similar days, the similar days can be evaluated by using the previous forecast day and its previous similar days.

The fuzzy inference system is used to evaluate the similarity between the previous forecast day and the previous similar days which results in the correction factors. The correction factors are used to correct the similar days of the forecast day to obtain the value of the load forecast. To evaluate the degree of the similarity, 3 fuzzy input variables are taken namely:

$$E_L^k = L_P - L_{ps}^k \tag{5}$$

$$E_T^k = T_p - T_{ps}^k \tag{6}$$

$$E_H^k = H_p - H_{ps}^k \tag{7}$$

Here, L_p and L_{ps} are the average load of the previous forecast and the previous kth similar day respectively. Similarly, T_p, T_{ps}, H_p, and H_{ps} are the corresponding variables for the temperature and humidity respectively. E_L, E_T and E_H take the fuzzy set values low, medium and high.

The fuzzy rules for the fuzzy inference system are based on the generalized knowledge of the effect of each of the variable on the load. If the membership of E_L is μ_{EL} that of E_T is μ_{ET} and E_H is μ_{EH} then the firing strength μ can be calculated using the min operator.

$$\mu = \min (\mu_{EL}^i, \mu_{ET}^i, \mu_{EH}^i) \tag{8}$$

The crisp value of the correction factors are then obtained by the centroid defuzzyfication scheme

$$W_K = \sum \alpha_i \, \mu_i^k / \sum \mu_i^k \tag{9}$$

W_k is the correction factor for the load curve on the kth similar day to shape the forecast day and the forecast curve for the next day is given by

$$L\,(t) = 1/N \left[\sum (1 + W_k) * L_s^k\,(t)\right] \tag{10}$$

Here, N is the number of the similar days taken and ‘t’ is hourly time from 1 to 24.

The forecast results deviation from the actual values are represented in the form of Mean absolute percentage error (MAPE) [5], [6]. It is defined as follows:

$$\text{MAPE} = (1/N) * (\sum (|P_A - P_F| / P_A)) \; *100 \tag{11}$$

Here, P_A and P_F are the actual and the forecast values of the load, and N is the number of hours of the day i.e. N = 1, 2, 3… 24.

3.3 Ant Colony Optimization

Ant colony optimization (ACO) is a paradigm for designing meta heuristic algorithms for combinatorial optimization problems. The main underlying idea, loosely inspired by the behavior of the real ants is that of a parallel search over several constructive computational threads based on local problem data and on a dynamic memory structure containing information on the quality of the previously obtained result.

A combinatorial optimization problem is a problem defined over a set C = c_1, c_2, …, c_n of the basic components. A subset S of the component represent the solution of the problem. F Subset of 2^c is a subset of feasible solutions, thus a solution S is feasible if and only if S belongs to F. A cost function Z is defined over the solution domain Z : 2c -> R, the objective being to find a minimum cost feasible solution S* i.e. to find S* : S* belongs to F and Z(S*) <= Z(S) for all S belongs To F.

A set of computational concurrent and asynchronous agents (a colony of ants) moves through the states of the problem corresponding to partial solutions of the problem to solve. They move by applying a stochastic local decision policy based on 2 parameters called as Trails and Attractiveness. By moving each ant incrementally constructs a solution to the problem. The ant is a simple computational agent, which iteratively constructs a solution for the instance to solve. Partial problems solutions are seen as states. Each ant moves (performs a step) from state ɩ to another one φ; corresponding to a more complete partial solution. At each step σ, each ant k computes a set of $A_k^{\sigma}(\iota)$ of feasible expansions to its current state and moves to one of these in probability [12].

For the ant k, the probability $p_{\iota\varphi}^{k}$ of moving from state ι to state φ depends on the combination of

1) Attractiveness (nɩφ)
2) Trail Level (τɩφ)

$$p_{\iota\psi}^{k} = \begin{cases} \dfrac{\tau_{\iota\psi}^{\alpha}+\eta_{\iota\psi}^{\beta}}{\sum_{(\iota\zeta)\notin \mathrm{tabu_k}}\left(\tau_{\iota\zeta}^{\alpha}+\eta_{\iota\zeta}^{\beta}\right)} & \text{if } (\iota\psi) \notin \mathrm{tabu_k} \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

The move probability distribution is defined by $p_{\iota\varphi}^{k}$ to be equal to zero for all moves which are infeasible for ant's k. The actual formula for the probability is given as below tabu_k is the tabu list of ant k. α and β are the impact of trail and the impact of attractiveness respectively.

After each iteration 't' of the algorithm when all ants have completed the solution trails are updated by means of the formula:

$$\tau_{\iota\varphi}^{(\tau)} = \rho\tau_{\iota\varphi}(\tau - 1) + \Delta\tau_{\iota\varphi}. \tag{13}$$

ρ is the evaporation coefficient and $\Delta\tau_{\iota\varphi}$ is the sum of contribution of all the ants. The value of the evaporation coefficient lies between 0 and 1 both included and it's a user defined coefficient.

The quality of the solution of the ant k would be the length L_k of the tour found by the ant and formula is given by

$$\tau_{ij}(t) = \rho\tau_{ij}(t\text{-}1) + \Delta\tau_{ij} \tag{14}$$

and

$$\Delta\tau_{ij} = \sum \Delta\tau_{ij}k \quad (k = 1..m) \tag{15}$$

Here, m is the number of the ants and $\Delta\tau_{ij}$ is the amount of trail laid on the edge (i, j) by ant k. We are using this algorithm to optimize the input parameters of the fuzzy system i.e. (a_1 to a_6).

3.4 Evaluation of the Algorithm

```
1. {Initialization} Initialize $\tau_{\iota\varphi}$ and $n_{\iota\varphi}$, for all $\iota\varphi$.
2. {construction} For each ant k (currently in state $\iota$)
   do repeat: choose in probability the state to move
   into append the chosen move to the $k^{th}$ ants set $tabu_k$
   until ant k has completed its solution.
3. {Trail update} for each ant move ($\iota\varphi$) do compute $\Delta\tau_{\iota\varphi}$
   update the trail matrix.
4. {Terminating condition} If not (end test) go to step 2.
```

4 Simulation Results

We use a real data set of load and weather variables collected over a fairly large period; to be precise it is a seven months data. The fuzzy inference system has been simulated using the fuzzy logic toolbox available in MATLAB and ant colony optimization algorithm has been implemented using programming in the MATLAB. The values of the parameters of the ant colony optimizations are $\alpha = 0.32$, $\beta = 0.59$, $\rho = 0.13$, $k = 100$.

Previous 30 days have been considered for selection of the 5 similar days (i.e., N = 5) for the forecasting. The fig. 4, fig. 5 and fig. 6 describe the membership functions for the load, temperature and the humidity respectively. For the short term load forecasting, temperature and humidity values for the day for which we want to forecast the load, are available to power engineers/load forecaster through weather forecasting data which comes through atmospheric models for weather forecasting. Therefore, the present work uses the forecasted values of temperature and humidity, procured from atmospheric forecasting model, for the day for which load is being forecasted. These variables belong to a three valued fuzzy set, namely Low (L), Medium (M) and High (H), as shown in figs. 4–6.

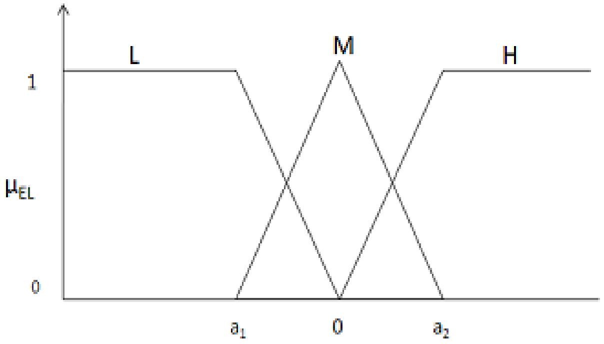


Fig. 4. Input membership for the load

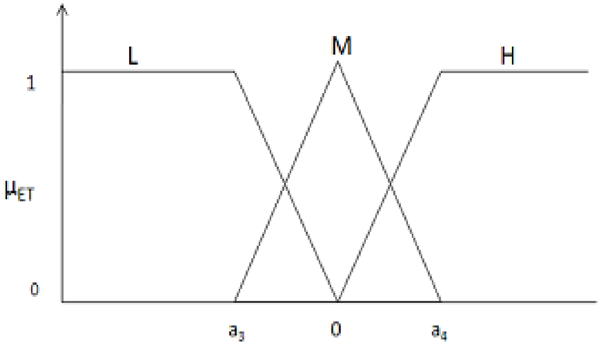


Fig. 5. Input membership for the temperature

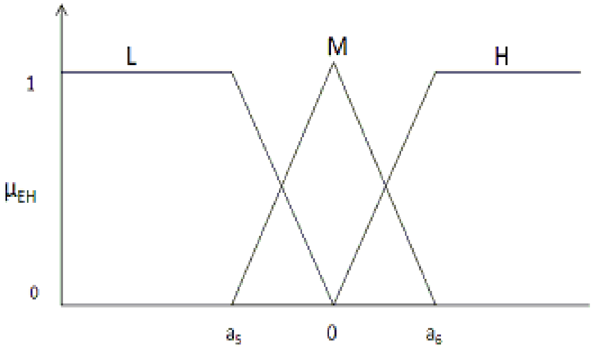


Fig. 6. Input membership for the humidity

Input parameters values to the fuzzy inference system, shown in figs. 4 – 6, are given in table 1 before and after the ant colony optimization.

Table 1. Input membership parameters before and after Ant Colony Optimization

	(a_1, a_2)	(a_3, a_4)	(a_5, a_6)
Before optimization	(-1200, 1200)	(-10, 10)	(-10, 10)
After Ant colony Optimization	(-15933, 16592)	(-43.92, 45.35)	(-14.54, 31.29)

Correction factors (W_k) are calculated for the fuzzy inference system using the fuzzy input parameters both before optimization and after optimization using ant colony optimization technique and these are further used for calculating the forecasted load. Then the MAPE are calculated for both cases, which are given in table 2.

Table 2. MAPE before and after Ant Colony Optimization

Days of forecast	MAPE before Ant colony optimization	MAPE after Ant colony optimization
24th July	1.48 %	0.96 %
25th July	1.76 %	1.38 %
26th July	2.55 %	2.07 %
27th July	2.02 %	1.12 %

The forecasted load curves for four typical days of a week, the 24th July, 25th July, 26th July and 27th July, are given in figs. 7-10.

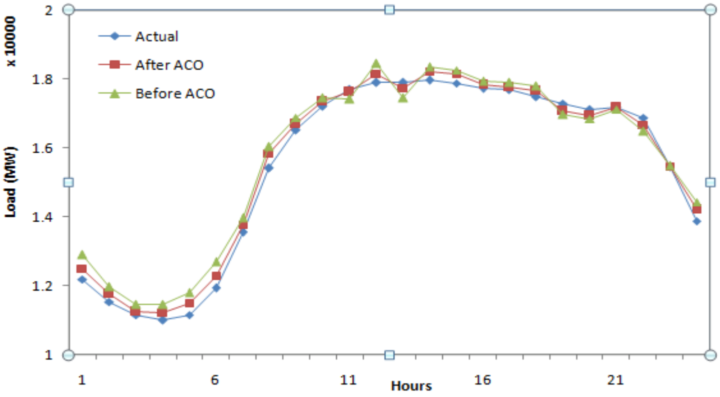


Fig. 7. The hourly Load forecast for 24th July

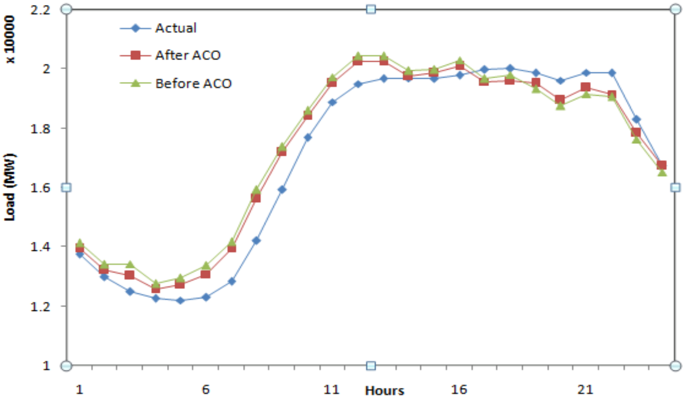


Fig. 8. The hourly Load forecast for 25th July

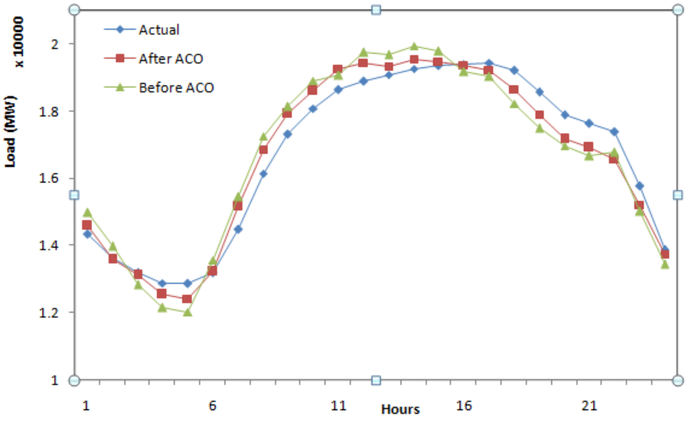


Fig. 9. The hourly Load forecast for 26th July

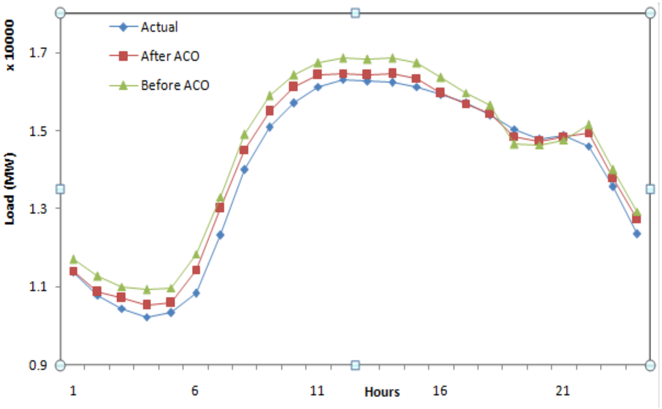


Fig. 10. The hourly Load forecast for 27th July

5 Conclusions

A short term load forecasting methodology using the fuzzy inference system followed by ant colony optimization is proposed in this paper which not only takes into account the value of the present temperature and humidity but also the values of the previous hour temperature and the humidity including the day type. A Euclidean norm with the weight factors is used to evaluate the degree of the similarity with that of the forecast day where previous 30 days are used. In the proposed method a fuzzy inference system is initially prepared and then that fuzzy inference system is optimized using ant colony optimization algorithm which further improves the results.

The results for the four representative days in the month of July are presented. The results obtained from the simulation gives good short term load forecasting data and they are within the range of 2.1 % MAPE. Moreover, this method can yield more accurate results if the historical data set is large including the previous year's data such that the better seasonal trend can also be accounted for.

In summary, we have proposed a methodology for short term load forecasting with fuzzy adaptive inference and the similarity along with ant colony optimization which uses temperature as well as humidity as the weather variables and which also provides the way to include more weather variables and other types of the optimization for the further research in the short term load forecasting process.

References

1. Rahman, S., Hazim, O.: A generalized knowledge-based short term load-forecasting technique. IEEE Trans. Power Syst. 8(2), 508–514 (1993)
2. Moghram, I., Rahman, S.: Analysis and evaluation of five short term load forecasting techniques. IEEE Transactions on Power Systems 4(4), 42–43 (1989)
3. Heinemann, G.T., Nordman, D.A., Plant, E.C.: The relationship between summer weather and summer loads- A regression analysis. IEEE Transactions Power Apparatus and Systems PAS-85(11), 1144–1154 (1996)
4. Papalexopoulos, A.D., Hesternberg, T.C.: A Regression based approach to short term load forecasting. IEEE Transactions on Power Systems 5(4), 1535–1550 (1990)
5. Rahman, S., Shrestha, G.: A priority vector based technique for load forecasting. IEEE Trans. Power Syst. 6(4), 1459–1464 (1993)
6. Rahman, S., Bhatnagar, R.: An Expert System based algorithm for short term load forecast. IEEE Transactions on Power Systems 3(2), 392–399 (1988)
7. Jain, A., Srinivas, E., Rauta, R.: Short term load forecasting using fuzzy adaptive inference and similarity. World Congress on Nature and Biologically Inspired Computing (NaBIC), 1743–1748 (2009)
8. Srinivas, E., Jain, A.: A Methodology for Short Term Load Forecasting Using Fuzzy Logic and Similarity. In: The National Conference on Advances in Computational Intelligence Applications in Power, Control, Signal Processing and Telecommunications, NCACI (2009)
9. Gross, G., Galiana, F.: Short term load forecasting. In: Proc. IEEE, Special Issue on Computers in Power System Operations, pp. 1558–1573 (1987)
10. Khaled, M., Naggar, E.L., Khaled, A., Rumaih, A.L.: Electric load forecasting using genetic based algorithm, optimal filter Estimator and least squares technique. Comparative Study World Academy of Science, Engineering and Technology, 134–142 (2005)
11. Bardran, I., Zayyat, H.E.L., Halsa, G.: Short Term and Medium Term load Forecasting for jordan's power systems. American Journal of Applied Sciences 5(7), 763–768 (2008)
12. Maniezzo, V., Gamberdella, L.M., Lungi, F.B.: Ant Colony optimization. In: New Optimization Techniques in Engineering, pp. 101–117. Springer, Heidelberg (2004)

The Use of Strategies of Normalized Correlation in the Ant-Based Clustering Algorithm

Arkadiusz Lewicki[1], Krzysztof Pancerz[1], and Ryszard Tadeusiewicz[2]

[1] University of Information Technology and Management in Rzeszów, Poland
{alewicki,kpancerz}@wsiz.rzeszow.pl
[2] AGH University of Science and Technology, Kraków, Poland
rtad@agh.edu.pl

Abstract. The article presents a new approach to the evaluation process associated with the modification of the ant-based clustering algorithm. The main aim of this study is to determine the degree of impact of the proposed changes on the results of the implemented clustering algorithm, whose task is not only to obtain the lowest intra-group variance, but also to self-determine the amount of target classes. These modifications concern both a different way of choosing the radius of perception considering the neighborhood of objects in a search decision space, as well as a use of a completely different metric than the Euclidean one for calculating the dissimilarity of objects based on the components including the normalized angular correlation of objects under consideration.

Keywords: ATTA, metaheuristics strategy, heuristic decision-making system, clustering data, data mining, unsupervised clustering.

1 Introduction

At time when a scientific and technological revolution has led to a breakthrough in the development of tools for transferring and presenting data, information and mechanisms of its exploration have become the primary objective of the research to develop not only new but also the continuous improvement of existing ways of exploratory data analysis. In this process, we can distinguish elements such as assessment of the possibility of grouping, clustering and validation of the obtained clusters. The first of these elements is to verify whether there are already some defined classes of objects in the set of data processed in the same space of possible observations. Only then we should apply the algorithm of grouping data by which we obtain the division of the set of objects such that the intergroup variance is as large as possible, while the intra-group variance is as small as possible. In this way, we aim to obtain the best coefficient of homogeneity of clusters created. This coefficient will be subject to the final stage of assessment validation.

The following algorithms are the most commonly used in data clustering: the EM algorithm, ISODATA, k-means, MVC and CLUSTER/2. However, these algorithms are characterized by a large number of parameters, among which there is the key parameter that defines the number of groups of objects that should be obtained at the

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 637–644, 2011.
© Springer-Verlag Berlin Heidelberg 2011

end of the running process. The study [1] has shown, that in many cases, clusters existing in the world describe the normal distribution. The random variables are associated with the density function (1):

$$f(x) = \frac{\exp\{-\frac{1}{2}(x-\mu)^T \sum{}^{-1} x-\mu\}}{(2\pi)^{\frac{D}{2}} \mid\Sigma\mid^{\frac{1}{2}}} \tag{1}$$

for Σ - the covariance matrix, μ - the center of a cluster, D – a number of dimensions of examples x. A number of classes in the available space is variable, especially for analysis in real time, when we talk about the process of continuous updating of data. Therefore, this situation eliminates the possibility of using well-known and elaborated methods of mathematical statistics. In view of this fact, the development of intelligent methods based on non-deterministic approaches is increasingly popular. An interesting solution in this area is the implementation of mechanisms belonging to the swarm intelligence, among which particularly promising results are obtained using ant-based algorithms [2,3,4,5,6]. The algorithm used here is based on a strategy of imitating the behavior of ants using a toroidal grid. The available set of data objects placed on the grid is subjected to cluster. These objects are arranged in such a way that elements belonging to the same class are in their immediate vicinity in the two-dimensional space, whereas differing from each other in the present array of features are apart. Conception defined in this way caused the creation of such algorithms as Saca, ATTA, AntClass, ACLUSTER or the de Castro algorithm [6,7,8,9,10,14]. These algorithms vary in the function of determining the probability of raising and dropping objects (as well as the local density function $f(x)$ on which they are dependent). They also vary in a strategy of the movement of agents and in the radius of agent perception, which directly affects the size of the neighborhood functions, essential for the calculation of probabilities. Another difference in approaches is implemented heterogeneity or homogeneity of the agents (ants), which affect the strategy associated with setting the algorithm parameters based on the memory of individuals in the population about the quality of their results received so far. Empirical studies conducted by the authors of publications as well as other researchers [7,11,12,13,14,15,16] have shown that, regardless of the structuring of data objects, the best results can be obtained by the Handl and Knowles [6] ATTA algorithm. It is a clustering algorithm based on the improved Deneubourg model [17], where the probability of raising an object met by an ant is equal to:

$$P_p = (\frac{k_p}{k_p + f})^2 \tag{2}$$

where k_p is the threshold parameter, while f represents the ratio of the number of objects encountered by the agent on his way and the maximum number of objects that can be found in an established neighborhood. The probability of dropping the element transported by an ant is described by the following formula:

$$P_d = \left(\frac{f}{k_d + f}\right)^2 \tag{3}$$

for k_d meaning also the threshold. This idea, in case of the probability of both raising and dropping the object, has been subordinated in the ATTA algorithm to the local density function $f(x_i)$, which can be written as:

$$f(x_i) = \begin{cases} \frac{1}{\delta^2} \sum_{y \in N_\chi(i)} (1 - \frac{d(o_i, o_j)}{\alpha}), \; when \;\; f(x_i) > 0 \wedge \forall_y (1 - \frac{d(o_i, o_j)}{\alpha}) > 0 \\ 0, \qquad\qquad otherwise \end{cases} \tag{4}$$

where criterion $f(x_i) > 0 \wedge \forall_y (1 - \frac{\| x - y \|}{\alpha}) > 0$ has been introduced to eliminate the possibility of dropping the object in the vicinity of δ elements significantly different from it. The value of increasing the probability of raising an object is described here by the equation:

$$P_p = \begin{cases} 1, \quad when \; f(x_i) \le 1 \\ \frac{1}{f(x_i)^2}, \quad otherwise \end{cases} \tag{5}$$

and the probability of dropping an object is described as:

$$P_d = \begin{cases} 1, \quad when \; f(x_i) \ge 1, \\ f(x_i)^4, \quad otherwise \end{cases} \tag{6}$$

This approach ensures that the operations of raising and dropping objects under consideration are deterministic, for a very little value of the density function in the first case and a very high value in the second case. It is connected with the process accelerating the formation of clusters, but only in areas which consist of a large number of objects with similar attribute vectors. Moreover, in the algorithm, the process of clustering has been further accelerated by the introduction of memory for each ant enabling it to store information about previously performed actions, i.e., about the modification of positions of objects being transported earlier.

2 The Approach

Despite many significant advantages such as no need to pre-determine the number of clusters, the algorithm described here has also several disadvantages. The main one is a lack of stability of the obtained solutions. This is reflected, especially when we examine the data space with a small cardinality. Moreover, in case of a large number of collections, but with their low frequencies, the use of this algorithm leads to a number of resulting groups smaller than the real one. Can we fix this? The suggestion to solve this problem, which was proposed by the authors of the publication and preceded by empirical studies, involves modifying the linear dependence of the correlation of the main parameters of the algorithm, which are: a function of dissimilarity of compared objects and the α parameter indirectly associated with it, and the ratio of the tested neighborhood. A correlation value of the parameters for *m* elements of the considered data set can be defined as:

$$r = \frac{\sum_{i=1}^{m}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{m}(x_i - \overline{x})^2 \sum_{i=1}^{m}(y_i - \overline{y})^2}} \tag{7}$$

which leads to the simplified form:

$$r = \frac{\frac{1}{n}\sum_{i=1}^{m} x_i y_i - \overline{xy}}{\sqrt{(\frac{1}{n}\sum_{i=1}^{m} x_i^2 - \overline{x}^2)(\frac{1}{n}\sum_{i=1}^{m} y_i^2 - \overline{y}^2)}} \tag{8}$$

where $\overline{x}, \overline{y}$ are appropriate average values defined as:

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{m} x_i\,, \quad \overline{y} = \frac{1}{n}\sum_{i=1}^{m} y_i \tag{9}$$

In the ant-based clustering algorithm proposed previously by the authors, a similarity measure of feature vectors of compared objects takes into account the Euclidean metric. Meanwhile, the disadvantage of this approach is not only omitting vector components with very small dimensions, but also a very large computational time. Therefore, considering the components represented by the feature vectors of the analyzed elements, there can be applied here the previously unexplored approach based on metrics associated with the Jacquard coefficient, where the similarity of objects o_i and o_j is represented by the corresponding vectors of features.

$$d(o_i, o_j) = 1 - \frac{\sum_{k=1}^{m} x_{ik} * x_{jk}}{\sum_{k=1}^{m} x_{ik}^2 + \sum_{k=1}^{m} x_{jk}^2 - \sum_{k=1}^{m} x_{ik} x_{jk}} \tag{10}$$

where x_{ik} and x_{jk} represent the *k-th* feature of the vectors of objects o_i and o_j, respectively. In view of the fact that, in the modified algorithm, the degree of dissimilarity of objects is taken into account, the formula must be converted to the form:

$$d(o_i, o_j) = 1 - \frac{1}{2}\left(\frac{\sum_{k=1}^{m} x_{ik} * x_{jk}}{\sum_{k=1}^{m} x_{ik}^2 + \sum_{k=1}^{m} x_{jk}^2 - \sum_{k=1}^{m} x_{ik} x_{jk}}\right) \tag{11}$$

Another proposed change concerns the factor that defines the size of the tested neighborhood of the available solutions. Observation made previously, demonstrated the fact that, with increasing radius of ant perception, the number of objects, for which the value of the dissimilarity function is calculated (11), also increases. Therefore, this factor may be designated as $d^2 * \delta^2$ for d which is the quotient of the smallest and the largest distance to the initialization size of the neighborhood. The formula defining the density function has the form:

$$f(x_i) = \begin{cases} d * \delta^2 \sum\limits_{y \in N_\chi(i)} (1 - \frac{d(o_i, o_j)}{\alpha}), & when \ f(x_i) > 0 \wedge \forall_y (1 - \frac{d(o_i, o_j)}{\alpha}) > 0 \\ 0, & otherwise \end{cases} \tag{12}$$

The introduced modification should decrease the importance of the distance between the considered objects with increasing the radius of perception and, dependent on it, the size of the area of studied solutions.

Evaluation tests of the proposed solution were carried out on three data sets (labeled with S1, S2 and S3). They contained 900, 400 and 90 objects, respectively, characterizing feature vectors which are attributes representing economic data on financial markets shared by WBK. Optimal values of the main parameters of the algorithm determined by manual tuning on the sample data model are presented in Table 1.

Table 1. The adopted values of the initialization parameters of the implemented algorithm

Parameters	**Value**
Number of ants	200
Memory size	30
Step size	100
α	0.73
δ	2

In order to evaluate the results, the authors of the article compared the proposed method with the original ATTA algorithm and the reference *k*-means algorithm. The quality of the process of aggregation was measured using parameters such as the Dunn index, rand and F-measure.

3 Results

The results of the assessment of the proposed algorithm, the ATTA algorithm and the *k*-means algorithm made with the usage of three indexes for examined data sets are presented in Table 2.

The results show clearly that the algorithm works well for smaller data sets, while for high-cardinality data the choice of *k*-means algorithm would be a qualitatively better approach. Nevertheless, the evaluation process demonstrated an advantage of the proposed modifications over the standard ATTA algorithm for each of the three test sets.

Table 2. Evaluation of the quality of the proposed algorithm compared to the ATTA algorithm and the *k*-means algorithm

Data set	Index	ATTA algorithm	K-means algorithm	The proposed algorithm
S1	Dunn index	1.031	**1.211**	1.116
S1	Rand	0.563	**0.578**	0.576
S1	F-measure	0.676	0.705	**0.706**
S2	Dunn index	0.752	**0.983**	0.708
S2	Rand	0.504	**0.560**	0.522
S2	F-measure	0.473	**0.678**	0.574
S3	Dunn index	1.227	0.770	**1.391**
S3	Rand	0.886	0.875	**0.889**
S3	F-measure	0.764	0.747	**0.774**

In order to determine the impact of the target strategy and the approach based on the standard ant-based clustering algorithm to the linear dependency of parameters of the algorithm, there were also made calculations of the value of the correlation coefficient formula described previously (7). These results are presented in Table 3.

Table 3. The impact of the proposed solution to the linear correlation of the main parameters of the algorithm

Algorithm	Data set	The correlation coefficient
ATTA	S1	0.632280
ATTA	S2	0.415771
ATTA	S3	0.486394
The proposed algorithm	S1	0.132473
The proposed algorithm	S2	0.106767
The proposed algorithm	S3	0.124522

An analysis of the results listed in Table 3 shows that the proposed approach resulted also in a significant reduction of the coefficient of linear correlation between the parameters of the algorithm, which, for the standard ATTA algorithm, had a noticeable impact because the value (which is up to 1.0) oscillated here within 0.41 to 0.63.

4 Conclusions

Modifications of the algorithm proposed by the authors, inspired by the behavior of ants, whose main advantage is the return of satisfactory results, despite the lack of information about the number of resulting classes, show that a solution based on the metric with the Jacquard factor and modification of a decisive factor for the vicinity of the tested space have influence on reducing the linear dependence of the main parameters of the algorithm. For the studied data sets, a solution found by the algorithm was also burdened with an error smaller than for the original algorithm and the *k*-means algorithm. Although experiments have shown that the *k*-means algorithm achieved significantly better results than the accepted method of solving a defined problem, it should be noted that, in case of a non-deterministic algorithm, we cannot talk about the knowledge of the available data structures, while its advantages can include independence from the dimensionality of data.

Therefore, in the next step in the near future, research should attempt to determine the quality of the proposed solutions for incomplete data sets and data sets containing clusters of an irregular shape. For this type of data, there can be proved that, in fact, the division into the appropriate clusters is not strongly dependent on the modified parameters of the algorithm, and a very important role may be played by a selection of fine-tuning strategies for the α coefficient. It seems to be probable that the case, when features describing the objects being compared are characterized by a different factor of importance, may be problematic. Therefore, further experiments should include the use of a metric taking into account the importance of a given attribute of the object under consideration, with the possibility of using measures of similarity for the features of various types.

Acknowledgements. This paper has been partially supported by the grant No. N N519 654540 from the National Science Centre in Poland.

References

1. Hore, P.: Distributed clustering for scaling classic algorithms, Theses and Dissertations, University of South Florida (2004)
2. Lewicki, A., Tadeusiewicz, R.: The recruitment and selection of staff problem with an Ant Colony System, Backgrounds and Applications. AISC, vol. 2. Springer, Heidelberg (2010)
3. Lewicki, A., Generalized non-extensive thermodynamics to the Ant Colony System, Information Systems Architecture and Technology, System Analysis Approach to the Design, Control and Decision Support, Wroclaw (2010)
4. Lewicki, A.: Non-Euclidean metric in multi-objective Ant Colony Optimization Algorithms, Information Systems Architecture and Technology, System Analysis Approach to the Design, Control and Decision Support, Wroclaw (2010)
5. Lewicki, A., Tadeusiewicz, R.: An autocatalytic emergence swarm algorithm in the decision-making task of managing the process of creation of intellectual capital. Springer, Heidelberg (2011)
6. Handl, J., Knowles, J., Dorigo, M.: Ant-based clustering and topographic mapping. Artif. Life 12(1) (2006)
7. Decastro, L., Von Zuben, F.: Recent Developments In Biologically Inspired Computing. Idea Group Publishing, Hershey (2004)
8. Mohamed, O., Sivakumar, R.: Ant-based Clustering Algorithms: A Brief Survey. International Journal of Computer Theory and Engineering 2(5) (October 2010)
9. Dorigo, M., Di Caro, G., Gambarella, L.: Ant Algorithms for Discrete Optimization. Artificial Life 5(3) (1999)
10. Azzag, H., Monmarché, N., Slimane, M., Venturini, G., Guinot, C.: AntTree: A new model for clustering with artificial ants. In: IEEE Congress on Evolutionary Computation, vol. 4, pp. 2642–2647. IEEE Press, Canberra (2003)

11. Scholes, S., Wilson, M., Sendova-Franks, A., Melhuish, C.: Comparisons in evolution and engineering: The collective intelligence of sorting. Adaptive Behavior - Animals, Animats, Software Agents, Robots, Adaptive Systems 12(3-4) (2004)
12. Sendova-Franks, A.: Brood sorting by ants: two phases and differential diffusion. Animal Behaviour (2004)
13. Boryczka, B.: Ant Clustering Algorithm, Intelligent Information Systems. Kluwer Academic Publishers (2008)
14. Abbass, H., Hoai, N., McKay, R.: AntTAG, A new method to compose computer using colonies of ants. In: Proceedings of the IEEE Congress on Evolutianory Computation, Honolulu, vol. 2 (2002)
15. Vizine, A., de Castro, L., Hruschka, E., Gudwin, R.: Towards improving clustering ants: An adaptive clustering algorithm. Informatica Journal 29 (2005)
16. Ouadfel, S., Batouche, M.: An Efficient Ant Algorithm for Swarm-based Image Clustering. Journal of Computer Science 3(3)
17. Deneubourg, J., Goss, S., Franks, N., Sendova-Franks, A., Detrain, C., Chretien, L.: The dynamics of collective sorting robot-like ants and ant-like robots. In: Proceedings of the First International Conference on Simulation of Adaptive Behavior: From Animals to Animats. MIT Press, Cambridge (1990)
18. Das, S., Abraham, A., Konar, A.: Metaheuristic Clustering. Springer, Heidelberg (2009)

Ant Based Clustering of Time Series Discrete Data – A Rough Set Approach

Krzysztof Pancerz[1], Arkadiusz Lewicki[1], and Ryszard Tadeusiewicz[2]

[1] University of Information Technology and Management in Rzeszów, Poland
{kpancerz,alewicki}@wsiz.rzeszow.pl
[2] AGH University of Science and Technology, Kraków, Poland
rtad@agh.edu.pl

Abstract. This paper focuses on clustering of time series discrete data. In time series data, each instance represents a different time step and the attributes give values associated with that time. In the presented approach, we consider discrete data, i.e., the set of values appearing in a time series is finite. For ant-based clustering, we use the algorithm based on the versions proposed by J. Deneubourg, E. Lumer and B. Faieta. As a similarity measure, the so-called consistency measure defined in terms of multistage decision transition systems is proposed. A decision on raising or dropping a given episode by the ant is made on the basis of a degree of consistency of that episode with the knowledge extracted from the neighboring episodes.

Keywords: ant based clustering, consistency measure, episodes, rough sets, time series.

1 Introduction

In data mining, we encounter a diversity of methods supporting data analysis which can be generally classified into two groups, called supervised and unsupervised learning, respectively (cf. [2]). One of unsupervised approaches investigated by us is clustering of data. In the paper, we consider the ant-based clustering algorithm based mainly on versions proposed by Deneubourg [4], Lumer and Faieta [9]. The ant-based clustering algorithm discovers automatically clusters in numerical data without prior knowledge of possible number of clusters. It is a very important feature in the preliminary step of data analysis. Clustering algorithms examine data to find groups (clusters) of items (objects, cases) which are similar to each other and dissimilar to the items belonging to other groups. The concept of dissimilarity or dual similarity is the essential component of any form of clustering that helps us to navigate through the data space and form clusters. We try to incorporate into the ant-based clustering algorithm a consistency measure defined for temporal data (see [12], [13]). This consistency measure plays a role of a similarity measure.

The consistency factor is defined in terms of multistage decision transition systems (MDTSs) proposed in [12] to describe transitions among states observed

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 645–653, 2011.
© Springer-Verlag Berlin Heidelberg 2011

in the given systems. If we are interested in sequences of changes of states, then we may represent such changes by means of polyadic transition relations over the sets of states. A given MDTS represents such a relation. Each object in MDTS is referred to as an episode. We can extend a given MDTS by adding new episodes to it. Next, we can answer an important question: "what is a consistency factor of a new episode added to MDTS with the knowledge included in MDTS?". A proper algorithm enabling us to answer this question is shown in Section 3.

We consider time series discrete data as the data to be clustered. In time series data, each instance represents a different time step and the attributes give values associated with that time. In the presented approach, we consider discrete data, i.e., the set of values appearing in a time series is finite. It follows from the approach to calculating the consistency measure. We assume that multistage decision transition systems include only discrete values of attributes. In case of continuous values, there is a need to define another relation between episodes.

In the literature, there have been proposed various algorithms combining clustering with fuzzy or rough set theory (cf. [3], [10], [11]). Our method differs from those proposed in the literature as we use a different way to compute a similarity measure (in our approach, it is based on the knowledge expressed by the rules determining transitions between states in time series data).

2 Basic Notions

A concept of an information system is one of the basic concepts of rough set theory. Information systems are used to represent some knowledge of elements of a universe of discourse. An *information system* is a pair $S = (U, A)$, where U is a set of *objects*, A is a set of *attributes*, i.e., $a : U \to V_a$ for $a \in A$, where V_a is called a value set of a. A *decision system* is a pair $S = (U, A)$, where $A = C \cup D$, $C \cap D = \emptyset$, and C is a set of *condition attributes*, D is a set of *decision attributes*. Any information (decision) system can be represented as a data table whose columns are labeled with attributes, rows are labeled with objects, and entries of the table are attribute values.

Let $S = (U, A)$ be an information system. Each subset $B \subseteq A$ of attributes determines an equivalence relation on U, called an *indiscernibility relation* $Ind(B)$, defined as $Ind(B) = \{(u, v) \in U \times U : \underset{a \in B}{\forall} a(u) = a(v)\}$. The equivalence class containing $u \in U$ is denoted by $[u]_B$.

Let $X \subseteq U$ and $B \subseteq A$. The *B-lower approximation* $\underline{B}X$ of X and the *B-upper approximation* $\overline{B}X$ of X are defined as $\underline{B}X = \{u \in U : [u]_B \subseteq X\}$ and $\overline{B}X = \{u \in U : [u]_B \cap X \neq \emptyset\}$, respectively.

A temporal information system is a kind of an information system $S = (U, A)$, with a set U of objects ordered in time, i.e., $U = \{u_t : t = 1, 2, \ldots, n\}$, where u_t is the object observed in time t. By a time window on S of the length λ in a point τ we understand an information system $S' = (U', A')$, where $U' = \{u_\tau, u_{\tau+1}, \ldots, u_{\tau+\lambda-1}\}$, $1 \leq \tau$, $\tau + \lambda - 1 \leq n$, and A' is a set of all attributes from A defined on the domain restricted to U'. The length λ of S' is defined as $\lambda = card(U')$. In the sequel, the set A' of all attributes in any time window

$S' = (U', A')$ on $S = (U, A)$ will be marked, for simplicity, with the same letter A like in S.

3 Consistency Measure

A dynamic information system proposed by Z. Suraj in [15] includes information about transitions between states observed in a given system. It means that the dynamics is expressed by a transition relation defined in a dynamic information system and the term of a dynamic information system should be understood in this sense. Here, we give some crucial notions concerning dynamic information systems.

A *multistage transition system* is a pair $MTS = (U, T)$, where U is a nonempty set of states and $T \subseteq U^k$ is a polyadic transition relation, where $k > 2$. A *multistage dynamic information system* is a tuple $MDIS = (U, A, T)$, where $S = (U, A)$ is an information system called the *underlying system* of $MDIS$ and $MTS = (U, T)$ is a multistage transition system. The underlying system includes states observed in a given system whereas a transition system describes transitions between these states.

Let $S = (U, \{a\})$ be a temporal information system representing a time series, where $U = \{u_t : t = 1, 2, \ldots, n\}$. On the basis of S, we can define a multistage transition system MTS, and next, a multistage dynamic information system $MDIS$. S is an underlying system for $MDIS$. For each $\tau = 1 \ldots n - \lambda + 1$, we create a time window $S' = (U', \{a\})$, where $U' = \{u_\tau, u_{\tau+1}, \ldots, u_{\tau+\lambda-1}\}$. We obtain $n - \lambda + 1$ time windows of the length λ. All sequences $(u_\tau, u_{\tau+1}, \ldots, u_{\tau+\lambda-1})$ of objects define a polyadic transition relation $T \subseteq U^\lambda$.

Each element of a multistage transition relation T in a multistage dynamic information system $MDIS = (U, A, T)$ is a sequence of states (from the set U), which can be referred to as an episode. The episode is, in fact, a fragment of a time series. This fragment has a fixed length.

Let $MDIS = (U, A, T)$ be a multistage dynamic information system, where $T \subseteq U^k$. Each element $(u^1, u^2, \ldots, u^k) \in T$, where $u^1, u^2, \ldots, u^k \in U$, is called an episode in $MDIS$.

We can use a suitable data table to represent a multistage transition system. Such a table will represent the so-called multistage decision transition system. Let $MTS = (U, T)$ be a multistage transition system. A *multistage decision transition system* is a pair $MDTS = (U_T, A^1 \cup A^2 \cup \ldots \cup A^k)$, where each $t \in U_T$ corresponds exactly to one element of the polyadic transition relation T whereas attributes from the set A^i determine values from the i-th domain of T, where $i = 1, 2, \ldots, k$. If we consider a time series described by one attribute, then each episode $t \in U_T$ in $MDTS = (U_T, A^1 \cup A^2 \cup \ldots \cup A^k)$, where $A^1 = \{a^1\}$, $A^2 = \{a^2\}$, ..., $A^k = \{a^k\}$, can be presented as a vector $\langle a^1(t), a^2(t), \ldots, a^k(t) \rangle$.

For a given multistage decision transition system, we can consider its elementary decision transition subsystems defined as follows. An *elementary decision transition subsystem* of a multistage decision transition system $MDTS = (U_T, A^1 \cup A^2 \cup \ldots \cup A^k)$ is a decision transition system $DTS(i, i+1) = (U_T, A^i \cup A^{i+1})$, where: $i \in \{1, ..., k-1\}$.

Now, we recall some important notions concerning extensions of multistage decision transition systems given in [12].

Any nontrivial extension of a given multistage decision transition system $MDTS = (U_T, A^1 \cup A^2 \cup \ldots \cup A^k)$ includes new episodes such that for each episode t^* we have $a(t^*) \in V_a$ for each $a \in (A^1 \cup A^2 \cup \ldots \cup A^k)$.

Let $DTS(i, i+1) = (U_T, A^i \cup A^{i+1})$ be the elementary decision transition subsystem. For each attribute $a \in A^i$ and the new episode t^*, we can transform $DTS(i, i+1)$ into the system with irrelevant values of attributes. If $a(t^*) \neq a(t)$, where $t \in U_T$, then we replace $a(t)$ by the value * (denoting an irrelevant value). This means that we create a new system for which appropriate sets of attribute values are extended by the value *. The transformed system can be treated as an incomplete system. Therefore, instead of an indiscernibility relation and equivalence classes, we use a characteristic relation and characteristic sets (cf. [6]). For the transformed elementary decision transition subsystem $DTS(i, i+1) = (U_T, A^i \cup A^{i+1})$, we define a characteristic relation $R(A^i)$. $R(A^i)$ is a binary relation on U_T defined as $R(A^i) = \{(t, v) \in U_T^2 : [\exists_{a \in A^i} a(t) \neq *] \wedge [\forall_{a \in A^i} (a(t) = * \vee a(t) = a(v))]\}$.

For each $t \in U_T$, a characteristic set $K_{A^i}(t)$ has the form $K_{A^i}(t) = \{v \in U_T : (t, v) \in R(A^i)\}$. For any $X \subseteq U_T$, the A^i-lower approximation of X is determined as $\underline{A^i}X = \{t \in U_T : K_{A^i}(t) \neq \emptyset \wedge K_{A^i}(t) \subseteq X\}$.

Let $DTS(i, i+1) = (U_T, A^i \cup A^{i+1})$ be an elementary decision transition subsystem, $a \in A^{i+1}$, and $v_a \in V_a$. By $X_a^{v_a}$ we denote the subset of U_T such that $X_a^{v_a} = \{t \in U_T : a(t) = v_a\}$.

For each episode t^* from the extension $MDTS$, we define a consistency factor of t^* (see [12], [13]). For a given multistage decision transition system $MDTS = (U_T, A^1 \cup A^2 \cup \ldots \cup A^k)$ we create a family **DTS** of elementary decision transition subsystems, i.e., $\mathbf{DTS} = \{DTS(i, i+1) = (U_T, A^i \cup A^{i+1})\}_{i=1,\ldots,k-1}$. Next, the consistency factor $\xi_{DTS(i,i+1)}(t^*)$ of the episode t^* with the knowledge included in $DTS(i, i+1)$ is computed for each subsystem $DTS(i, i+1)$ from the family **DTS**. Finally, the consistency factor $\xi_{MDTS}(t^*)$ of the episode t^* with the knowledge included in $MDTS$ is calculated as:

$$\xi_{MDTS}(t^*) = \mathop{O}_{i=1}^{k-1} \xi_{DTS(i,i+1)}(t^*),$$

where O denotes an aggregation operator. The consistency factor $\xi_{DTS(i,i+1)}(t^*)$ of the episode t^* with the knowledge included in $DTS(i, i+1)$ is calculated as:

$$\xi_{DTS(i,i+1)}(t^*) = 1 - \frac{card(\widetilde{U}_T)}{card(U_T)},$$

where:

$$\widetilde{U}_T = \bigcup_{a \in A^{i+1}} \bigcup_{v_a \in V_a} \{\underline{A^i}X_a^{v_a} : \underline{A^i}X_a^{v_a} \neq \emptyset \wedge a(t^*) \neq v_a\}.$$

A consistency factor may be calculated using Algorithm 1. In this algorithm, we assume an arithmetic average as an aggregation operator.

Algorithm 1. Algorithm for computing a consistency factor of an episode belonging to the extension of a given multistage decision transition system

```
Input  : A multistage decision transition system
         $MDTS = (U_T, A^1 \cup A^2 \cup \ldots \cup A^k)$, an episode $t^*$ from any extension
         of $MDTS$.
Output : A consistency factor $\xi_{MDTS}(t^*)$ of the episode $t^*$ with the knowledge
         included in $MDTS$.
$s \longleftarrow 0$;
for each elementary decision transition subsystem
$DTS(i, i+1) = (U_T, A^i \cup A^{i+1})$ of $MDTS$ do
    Compute $\xi_{DTS(i,i+1)}(t^*)$ using Algorithm 3 (treat $DTS(i, i+1)$ as a
    decision system and $t^*$ as an object added to the decision system);
    $s \longleftarrow s + \xi_{DTS(i,i+1)}(t^*)$;
end
$\xi_{MDTS}(t^*) \longleftarrow \frac{s}{k-1}$
```

Algorithm 2. Algorithm for computing a consistency factor of an object belonging to the extension of a decision system

```
Input  : A decision system $S = (U, A \cup D)$, a new object $u^*$ added to $S$.
Output : A consistency factor $\xi_S(u^*)$ of the object $u^*$ with the knowledge
         included in $S$.
$\widetilde{U} \longleftarrow \emptyset$;
for each $d \in D$ do
    Create a copy $S' = (U, A \cup D)$ of $S$;
    for each $u \in U$ in $S'$ do
        for each $a \in A$ in $S'$ do
            if $a(u) \neq a(u^*)$ then
                $a(u) \longleftarrow *$;
            end
        end
    end
    Remove each object $u \in U$ in $S'$ such that $\underset{a \in A}{\forall} a(u) = *$;
    for each $v_d \in V_d$ do
        $X_d^{v_d} \longleftarrow \{u \in U : d(u) = v_d\}$;
        if $\underline{C}X_d^{v_d} \neq \emptyset$ then
            if $d(u^*) \neq v_d$ then
                $\widetilde{U} \longleftarrow \widetilde{U} \cup \underline{C}X_d^{v_d}$;
            end
        end
    end
end
$\xi_S(u^*) \longleftarrow 1 - \frac{card(\widetilde{U})}{card(U)}$;
```

Algorithm 3. Algorithm for Ant Based Clustering

```
for each episode $t \in U_T$ do
    Place $t$ randomly on a grid $G$;
end
for each ant $a_j \in Ants$ do
    Place $a_j$ randomly on a grid place occupied by one of episodes from $U_T$;
end
for $k \leftarrow 1$ to $n_{it}$ do
    for each ant $a_j \in Ants$ do
        if $a_j$ is unladen then
            if place of $a_j$ is occupied by dropped episode $t$ then
                Draw a random real number $r \in [0, 1]$;
                if $r \leq p_{pick}(t)$ then
                    set $t$ as picked;
                    set $a_j$ as carrying the episode;
                end
            else
                move $a_j$ randomly to another place occupied by one of episodes
                from $U_T$;
            end
        else
            Draw a random real number $r \in [0, 1]$;
            if $r \leq p_{drop}(t)$ then
                move $t$ randomly to a new place on a grid;
                set $t$ as dropped;
                set $a_j$ as unladen;
            end
        end
    end
end
```

4 Ant Based Clustering

The ant based clustering algorithm used in our experiment is mainly based on the versions proposed by Deneubourg [4], Lumer and Faieta [9]. There has been introduced a number of slight modifications that adjust the algorithm to the considered problem. For more details, the reader is referred to papers devoted to different studies on features and modifications of Lumer and Faieta's algorithm for data clustering (for example, see [1], [5], [7], [8]). Here, we present only the most significant information. Algorithm 3 may be used to cluster episodes appearing in a given time series. Let $MDTS = (U_T, A^1 \cup A^2 \cup \ldots \cup A^k)$ be a multistage decision transition system representing episodes in a time series. U_T is a set of episodes being clustered (the size of U_T is n), n_{it} is a number of iterations performed for the clustering process, *Ants* is a set of ants used in the clustering process, $p_{pick}(t)$ and $p_{drop}(t)$ are probabilities for the picking and dropping operations made for a given episode t, respectively (see Formulas 1 and 2).

For picking and dropping decisions the following threshold formulas are used

$$p_{pick}(t) = \begin{cases} \frac{1-f_{loc}(t)}{1+f_{loc}(t)} & \text{if } f_{loc}(t) < 0.95 \\ 0.95 & otherwise \end{cases} \tag{1}$$

and

$$p_{drop}(t) = f_{loc}^4(t) \tag{2}$$

respectively, where f_{loc} is a neighborhood function. $f_{loc}(t) = \xi_{MDTS^*}(t)$, where $MDTS^*$ includes only those episodes from $MDTS$ which are placed in a local neighborhood of t. This local neighborhood is built from the cells on a grid surrounding locally the cell occupied by the episode t.

5 Experiments

Basic experiments validating the proposed approach were carried out on artificial data. We have clustered different time series (episodes) of three shapes: sinusoidal, cosinusoidal and exponential. Time series have been transformed into the so-called delta representation, i.e., attribute values have been replaced with differences between current values and values of previous instances. After transformation, each time series was a sequence consisting of three values: -1 (denoting decreasing), 0 (denoting a lack of change), 1 (denoting increasing). In this case, we have obtained multistage decision transition systems with discrete values. Episodes have been clustered on two-dimensional grid of size 50. An exemplary result of clustering is shown in Figure 1. Crosses correspond to sinusoidal episodes, diamonds correspond to exponential episodes, and circles correspond to cosinusoidal episodes.

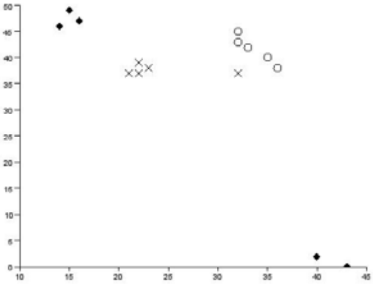


Fig. 1. An exemplary result of clustering

6 Conclusions

In the paper, we have examined a problem of clustering time series discrete data using the ant based approach. As a similarity measure, a consistency measure, defined in terms of multistage decision transition systems, has been used. In the future, we will apply the proposed approach to real-life data. Moreover, it seems to be necessary to tune a consistency measure to the considered problem.

Acknowledgments. This paper has been partially supported by the grant No. N N519 654540 from the National Science Centre in Poland.

References

1. Boryczka, U.: Finding groups in data: Cluster analysis with ants. Applied Soft Computing 9(1), 61–70 (2009)
2. Cios, K., Pedrycz, W., Swiniarski, R.W., Kurgan, L.: Data mining. A knowledge discovery approach. Springer, New York (2007)
3. Das, S., Abraham, A., Konar, A.: Metaheuristic Clustering. Springer, Heidelberg (2009)
4. Deneubourg, J., Goss, S., Franks, N., Sendova-Franks, A., Detrain, C., Chrétien, L.: The dynamics of collective sorting: Robot-like ants and ant-like robots. In: Proceedings of the First International Conference on Simulation of Adaptive Behaviour: From Animals to Animats, vol. 1, pp. 356–365. MIT Press, Cambridge (1991)
5. Gilner, B.: A comparative study of ant clustering algorithms (2007)
6. Grzymała-Busse, J.W.: Data with Missing Attribute Values: Generalization of Indiscernibility Relation and Rule Induction. In: Peters, J.F., Skowron, A., Grzymała-Busse, J.W., Kostek, B.z., Świniarski, R.W., Szczuka, M.S. (eds.) Transactions on Rough Sets I. LNCS, vol. 3100, pp. 78–95. Springer, Heidelberg (2004)
7. Handl, J., Knowles, J., Dorigo, M.: Ant-based clustering and topographic mapping. Artificial Life 12(1), 35–62 (2006)
8. Handl, J., Meyer, B.: Ant-based and swarm-based clustering. Swarm Intelligence 1, 95–113 (2007)
9. Lumer, E., Faieta, B.: Diversity and adaptation in populations of clustering ants. In: Proceedings of the Third International Conference on Simulation of Adaptive Behaviour: From Animals to Animats, vol. 3, pp. 501–508. MIT Press, Cambridge (1994)
10. Mitra, S., Banka, H., Pedrycz, W.: Rough-fuzzy collaborative clustering. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 36, 795–805 (2006)
11. Mitra, S., Pedrycz, W., Barman, B.: Shadowed c-means: Integrating fuzzy and rough clustering. Pattern Recognition 43, 1282–1291 (2010)
12. Pancerz, K.: Extensions of dynamic information systems in state prediction problems: the first study. In: Magdalena, L., Ojeda-Aciego, M., Verdegay, L. (eds.) Proceedings of the 12th International Conference on Information Processing and Management of Uncertainty in Knowledge-based Systems (IPMU 2008), Malaga, Spain, pp. 101–108 (2008)

13. Pancerz, K.: Extensions of Multistage Decision Transition Systems: The Rough Set Perspective. In: Cyran, K.A., Kozielski, S., Peters, J.F., Stańczyk, U., Wakulicz-Deja, A. (eds.) Man-Machine Interactions. AISC, vol. 59, pp. 209–216. Springer, Heidelberg (2009)
14. Pawlak, Z.: Rough Sets. Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht (1991)
15. Suraj, Z.: The Synthesis Problem of Concurrent Systems Specified by Dynamic Information Systems. In: Polkowski, L., Skowron, A. (eds.) Rough Sets in Knowledge Discovery 2. Applications, Case Studies and Software Systems, pp. 418–448. Physica-Verlag, Heidelberg (1998)

Sensor Deployment for Probabilistic Target *k*-Coverage Using Artificial Bee Colony Algorithm

S. Mini[1], Siba K. Udgata[1], and Samrat L. Sabat[2]

[1] Department of Computer and Information Sciences
University of Hyderabad, Hyderabad-500046, India
`mini2min2002@yahoo.co.in, udgatacs@uohyd.ernet.in`
[2] School of Physics
University of Hyderabad, Hyderabad-500046, India
`slssp@uohyd.ernet.in`

Abstract. A higher level of coverage is required for many sensitive applications. Though initial work on target coverage problems in wireless sensor networks used binary sensing model, a more realistic sensing model, the probabilistic sensing model has been used later. This work considers probabilistic k-coverage; where the required level of coverage has to be satisfied with k sensors and each target should also be monitored with a specific probability. We compute the optimal deployment location of sensor nodes, such that the probabilistic coverage as well as the k-coverage requirement is satisfied with the required sensing range being optimal. Preliminary results of using artificial bee colony algorithm to solve deployment problem for probabilistic target k-coverage is reported in this paper.

Keywords: Sensor Deployment, Probabilistic Coverage, Target Coverage, ABC Algorithm, k-coverage.

1 Introduction

Development of an energy-efficient wireless sensor network introduces many challenges. Coverage problem is one such issue. The widely addressed coverage problems are area coverage and target coverage. Area coverage concerns with how well a region is monitored and target coverage deals with how well point objects(targets) in the region are monitored. Since sensor nodes are battery powered, energy should be properly used to extend the network lifetime.

Deployment of sensor nodes can be either random or deterministic. For a random deployment, sensor nodes can be scheduled in such a way that all nodes need not be active at the same time for coverage requirement to be satisfied. If the nodes can be deterministically deployed and if the targets are fairly large in number compared to the given number of sensor nodes, energy usage can be restricted by limiting the sensing range requirement.

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 654–661, 2011.
© Springer-Verlag Berlin Heidelberg 2011

Although work has been done in area coverage and target coverage problems, the results have been reported either for a binary detection model or for a probabilistic sensing model. In this paper, we address probabilistic target k-coverage for deterministic deployment of sensor nodes. The target is either monitored with full confidence or not monitored in a binary sensing model. With probabilistic model, the probability that the sensor detects a target depends on the relative position of the target within the sensors' sensing range.

Probabilistic coverage applies with some kinds of sensors e.g. acoustic, seismic etc., where the signal strength decays with distance from the source, and not with sensors that only measure local point values e.g. temperature, humidity, light etc.[1]. It can be used for applications which require a certain degree of confidence.

In probabilistic k-coverage, given a region with n sensor nodes and m targets, each target needs to be monitored by at least k sensors, where $1 \leq k \leq n$, as well as monitored with a required probability p for the network to function. We use Artificial Bee Colony algorithm [2][3][4] to compute the optimal deployment locations so that the sensing range requirement is at minimum, satisfying k-coverage and probabilistic coverage.

The rest of the paper is organized as follows: Section 2 presents an overview of related work. In Section 3, the problem is defined. We present the proposed method in Section 4. The proposed method is evaluated by simulations in Section 5. Section 6 concludes the paper.

2 Related Work

Ahmed et al. [1] propose a probabilistic coverage algorithm to evaluate area coverage in a randomly deployed wireless sensor network. The coverage issues in wireless sensor networks are investigated based on probabilistic coverage and propose a distributed Probabilistic Coverage Algorithm (PCA) to evaluate the degree of confidence in detection probability provided by a randomly deployed sensor network. Zou et al. [5] address the problem of selecting a subset of nodes that are active for both sensing and communication. The active node selection procedure is aimed at providing the highest possible coverage of the sensor field, i.e., the surveillance area. It also assures network connectivity for routing and information dissemination.

Carter et al. [6] address area coverage problem for probabilistic deployment of sensor nodes. The problem is formulated as an optimization problem with the objective to minimize cost while covering all points above a given probability of detection coverage threshold. A probabilistic coverage matrix is defined to decouple the coverage determination method from the model. A GA (Genetic Algorithm) approach is presented to solve the optimization problem. Hefeeda et al. [7] propose and evaluate a fully distributed, probabilistic coverage protocol. Experimental study shows that the protocol activates less sensors than the others while maintaining the same level of coverage.

Udgata et al. [8] use artificial bee colony algorithm to solve sensor deployment problem in irregular terrain. This addresses area coverage problem where the area under consideration is an irregular terrain. Mini et al. [9] solve sensor deployment problem for simple coverage problem in 3-D terrain using artificial bee colony algorithm where the optimal deployment position such that the required sensing range is minimum for each target to be monitored by at least one sensor node, is computed. Mini et al. [10] use artificial bee colony algorithm to solve sensor deployment problem for simple,k and Q-coverage problems in 3-D terrain. For simple, k and Q-coverage, the optimal deployment position where the required sensing range is minimum is identified. [8][9][10] focus on solving deployment problem in wireless sensor network using binary sensing model. Unlike all the above works, in this paper, we address sensor deployment problem for probabilistic k-coverage.

3 Problem Definition

Let S $= \{S_1, S_2, \ldots, S_n\}$ be the set of sensor nodes and T $= \{T_1, T_2, \ldots, T_m\}$ be the set of targets in a given region. A sensor node located at (x_1, y_1, z_1) can cover a target at (x_2, y_2, z_2) if the euclidean distance between the sensor node and the target is less than or equal to the sensing range r.

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \leq r \tag{1}$$

3.1 Binary Sensing Model

A binary sensing model is given by,

$$M_{ij} = \begin{cases} 1 & \text{if } d_{ij} \leq r, \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$. d_{ij} corresponds to the euclidean distance between T_i and S_j

3.2 Probabilistic Coverage

As in [5], we use the following exponential function to represent the confidence level in the received sensing signal:

$$M_{ij} = \begin{cases} e^{-\alpha d_{ij}} & \text{if } d_{ij} \leq r, \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where $0 \leq \alpha \leq 1$ is a parameter representing the physical characteristics of the sensing unit and environment.

The coverage of a target T_i which is monitored by multiple sensor nodes $\mathbf{S}_i$ is given by,

$$M_i(\mathbf{S}_i) = 1 - \prod_{S_j \in \mathbf{S}_i} (1 - M_{ij}) \tag{4}$$

3.3 *k*-Coverage

Given a set of targets T $= \{T_1, T_2, \ldots, T_m\}$ located in a region and a set of sensor nodes S $= \{S_1, S_2, \ldots, S_n\}$, each target in T should be monitored by at least k number of sensor nodes, $1 \leq k \leq n$.

3.4 Mean of Location Points

The mean value of the location points (x_q, y_q, z_q) for $q = 1, 2, \cdots N$, is represented by (a_1, a_2, a_3), where

$$a_1 = \frac{\sum_{q=1}^{N}(x_q)}{N} \tag{5}$$

$$a_2 = \frac{\sum_{q=1}^{N}(y_q)}{N} \tag{6}$$

$$a_3 = \frac{\sum_{q=1}^{N}(z_q)}{N} \tag{7}$$

3.5 Sensor Deployment to Achieve Probabilistic Target *k*-Coverage

Given a set of m targets $T = \{T_1, T_2, \ldots, T_m\}$ located in $u \times v \times w$ region and n sensor nodes $S = \{S_1, S_2, \ldots, S_n\}$, place the nodes such that each target is monitored by at least k-sensor nodes and with a total probability p and sensing range is minimum. The objective is to cover each target with at least k sensor nodes and probability p and to minimize the function

$$F = \forall_j((max(distance(S_j, H_g)))) \tag{8}$$

where H is the set of all targets monitored by S_j, $j = 1, 2, \ldots, n$, $g = 1, 2, \ldots, h$, where h is the total number of targets S_j monitors.

4 Proposed Method

Let the solution population be B. The region is assumed to have only stationary targets. Each solution $B_e = \{(x_1, y_1, z_1), (x_2, y_2, z_2), \ldots, (x_n, y_n, z_n)\}$ where $e = 1, 2, \ldots, nb$, nb the total number of bees and n the total number of nodes to be deployed, corresponds to a bee. The initial solution is generated in such a way that all the targets can be probabilistically covered, and no sensor node is left idle without contributing to probabilistic k-coverage. The sensor nodes which can make each target T_i meet the required probability is then identified. Let this subset be R_i. If R_i satisfies k-coverage requirement of T_i, T_i is assigned to each sensor node in R_i. If it does not satisfy k-coverage, then identify nearest nodes which do not belong to R_i that can make T_i k-covered, along with R_i. T_i is assigned to these new sensor nodes in addition to R_i. Each sensor node is then moved to the center of all the targets which are assigned to it. This move is not

Algorithm 1. Proposed Scheme

```
Initialize the solution population $B$
Evaluate fitness
Produce new solutions based on probabilistic coverage and $k$-coverage
Choose the fittest bee
$cycle = 1$
repeat
    Search for new solutions in the neighborhood
    if new solution better than old solution then
        Memorize new solution and discard old solution
    end if
    Replace the discarded solution with a new randomly generated solution
    Memorize the best solution
    $cycle = cycle + 1$
until cycle = maximumcycles
```

allowed if some target will not be probabilistically covered due to this shift of location.

The Euclidean distance between each target and the sensor location to which it is associated is used as the fitness function to evaluate the solutions. Each sensor node is associated to a cluster, where a cluster corresponds to the set of targets monitored by the sensor node. Let $D_j = (D_{j1}, D_{j2}, D_{j3})$ be the initial position of j^{th} cluster. $F(D_j)$ refers to the nectar amount at food source located at D_j. After watching the waggle dance of employed bees, an onlooker goes to the region of D_j with probability P_j defined as,

$$P_j = \frac{F(D_j)}{\sum_{l=1}^{n} F(D_l)} \tag{9}$$

where n is the total number of food sources. The onlooker finds a neighborhood food source in the vicinity of D_j by using,

$$D_j(t+1) = D_j(t) + \delta_{ji} \times v \tag{10}$$

where δ_{ji} is the neighborhood patch size for i^{th} dimension of j^{th} food source, v is random uniform variate $\in$ [-1, 1] and calculates the fitness value. It should be noted that the solutions are not allowed to move beyond the edge of the region. The new solutions are also evaluated by the fitness function. If any new solution is better than the existing one, choose that solution and discard the old one. Scout bees search for a random feasible solution. The solution with the least sensing range is finally chosen as the best solution. Algorithm 1. explains the proposed scheme.

5 Results and Discussion

Initially, we consider a $10 \times 10 \times 10$ grid for experiments. The number of targets is 10 and the number of sensor nodes is 5. The number of bees is taken as 10,

Table 1. Sensing Range for Probabilistic Coverage

		Sensing Range		
α	Probability	Best	Mean	Standard Deviation
0.05	0.6	2.0616	2.0616	0
	0.7	2.0616	2.0616	0
	0.8	2.0616	2.0616	0
	0.9	2.0616	2.0616	0
0.1	0.6	2.0616	2.0616	0
	0.7	2.0616	2.0616	0
	0.8	2.0616	2.0616	0
	0.9	3.8748	3.9071	0.0465
0.15	0.6	2.0616	2.0616	0
	0.7	2.0616	2.0616	0
	0.8	3.8586	4.0558	0.3406
0.2	0.6	2.0616	2.0616	0
	0.7	3.5618	3.6170	0.0927

number of cycles is 500, limit for neighborhood search is 20 and the number of runs is 3. MATLAB 2007a is used for implementation.

5.1 Probabilistic Coverage

Initially, we compute the sensing range required for probabilistic coverage without considering k. The required probability for coverage is varied from to 0.6 to 0.9. α is varied from 0.05 to 0.2. Table 1. shows the sensing range requirement for probabilistic coverage. When $\alpha = 0.05$, the required sensing range does not change for any of the required probability. But when α increases to 0.1, the sensing range required increases for a detection probability of 0.9. Similarly, a variation in sensing range is observed at 0.8 for $\alpha = 0.15$ and at 0.7 for $\alpha = 0.2$. This implies that for higher α, the sensing range requirement varies at a smaller detection probability.

5.2 Probabilistic *k*-Coverage

To observe the difference in sensing range required for probabilistic k-coverage, k takes values 2 and 3. The required probability for coverage is varied from 0.6 to 0.9 and α is varied from 0.05 to 0.2.

Table 2. shows the sensing range requirement for probabilistic k-coverage. For a constant detection probability, the sensing range requirement may or may not increase with k. This is because for some cases, more than k sensor nodes may have to monitor a target for satisfying probabilistic coverage. Due to the same reason, there are instances where probabilistic coverage and probabilistic k-coverage requires the same sensing range. For example, $\alpha = 0.05$ and probability $= 0.6$ for probabilistic coverage requires sensing range of 2.0616 units, and $\alpha = 0.05$, probability $= 0.6$, $k = 1$ for probabilistic k-coverage requires the same sensing range. But in this case, when $k = 2$, the sensing range required increases.

We also consider a $100 \times 100 \times 20$ grid for experimentation. Three instances of 100 targets being monitored by 10 sensor nodes are considered. k is varied from

Table 2. Sensing Range for Probabilistic Target k-Coverage

			Sensing Range		
α	Probability	k	Best	Mean	Standard Deviation
0.05	0.6	2	3.9249	3.9844	0.0531
		3	6.1847	6.2085	0.0315
	0.7	2	3.7814	3.9384	0.1550
		3	6.1974	6.2051	0.0089
	0.8	2	3.9514	3.9802	0.0251
		3	6.1901	6.2034	0.0142
	0.9	2	3.8407	3.9497	0.1330
		3	6.1847	6.2054	0.0187
0.1	0.6	2	3.9175	3.9870	0.1050
		3	6.1847	6.2235	0.0366
	0.7	2	3.9098	3.9505	0.0423
		3	6.2043	6.2235	0.0262
	0.8	2	3.9105	3.9433	0.0289
		3	6.1870	6.2013	0.0125
	0.9	2	3.9206	4.2680	0.3030
		3	6.1968	6.2175	0.0319
0.15	0.6	2	3.9538	3.9658	0.0107
		3	6.1847	6.2187	0.0296
	0.7	2	3.9236	3.9490	0.0304
		3	6.1847	6.2129	0.0289
	0.8	2	3.9286	4.0131	0.0757
		3	6.2005	6.2144	0.0128
0.2	0.6	2	3.9319	3.9531	0.0198
		3	6.1851	6.1876	0.0042
	0.7	2	3.9119	3.9789	0.0749
		3	6.1847	6.1963	0.0113

Table 3. Experimental Results

		Sensing Range		
k	Instance	Best	Mean	Standard Deviation
1	1	19.2000	19.3030	0.2485
	2	19.2428	19.3317	0.1487
	3	18.6335	19.0473	0.3584
2	1	28.3267	28.5713	0.2630
	2	29.5044	29.7045	0.3058
	3	28.3521	28.5780	0.1957
3	1	41.8409	42.5757	0.6741
	2	43.9295	44.0164	0.1469
	3	43.7273	44.0340	0.5125
4	1	49.8290	49.8973	0.0611
	2	50.3353	50.5659	0.2514
	3	50.7494	51.2090	0.7510
5	1	51.6985	52.2129	0.4852
	2	52.192	52.498	0.3183
	3	54.0340	54.4637	0.5567

1 to 5. The required probability is set to 0.8 and α is assumed to be 0.01. Table 3. shows the sensing range requirement for this set-up. There is no significant variation in standard deviation even with an increase in k. This shows that the method is a reliable one even for higher k.

6 Conclusion and Future Work

This paper explores the use of both probabilistic coverage and probabilistic k-coverage models for target coverage problem and proposes a method to compute the optimal deployment locations so that the sensing range requirement is minimum. ABC algorithm has been used to solve both probabilistic coverage and probabilistic k-coverage. The variation in sensing range is studied for a range of detection probabilites (p), coverage requirement (k) and physical medium characteristics (α). ABC algorithm proves to be reliable in getting the optimal deployment locations. The standard deviation of obtained sensing range among various runs does not change significantly for a larger region or for higher values of k. In future, we plan to extend this work for probabilistic Q-coverage.

References

1. Ahmed, N., Kanhere, S.S., Jha, S.: Probabilistic Coverage in Wireless Sensor Networks. In: Proc. of IEEE LCN 2005, pp. 672–681 (2005)
2. Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. Journal of Global Optimization 39, 459–471 (2007)
3. Karaboga, D., Basturk, B.: On the performance of artificial bee colony (ABC) algorithm. Applied Soft Computing 8, 687–697 (2008)
4. Karaboga, D., Akay, B.: A survey: algorithms simulating bee swarm intelligence. Artificial Intelligence Review 31, 61–85 (2009)
5. Zou, Y., Chakrabarty, K.: A Distributed Coverage- and Connectivity-Centric Technique for Selecting Active Nodes in Wireless Sensor Networks. IEEE Transactions on Computers 54(8), 978–991 (2005)
6. Carter, B., Ragade, R.: A probabilistic model for the deployment of sensors. In: Sensors Applications Symposium, pp. 7–12 (2009)
7. Hefeeda, M., Ahmadi, H.: Energy-Efficient Protocol for Deterministic and Probabilistic Coverage in Sensor Networks. IEEE Transactions on Parallel and Distributed Systems 21(5), 579–593 (2010)
8. Udgata, S.K., Sabat, S.L., Mini, S.: Sensor Deployment in Irregular Terrain using ABC Algorithm. In: Proc. of IEEE BICA 2009, pp. 296–300 (2009)
9. Mini, S., Udgata, S.K., Sabat, S.L.: Sensor Deployment in 3-D Terrain using Artificial Bee Colony Algorithm. In: Panigrahi, B.K., Das, S., Suganthan, P.N., Dash, S.S. (eds.) SEMCCO 2010. LNCS, vol. 6466, pp. 424–431. Springer, Heidelberg (2010)
10. Mini, S., Udgata, S.K., Sabat, S.L.: Artificial Bee Colony based Sensor Deployment Algorithm for Target Coverage Problem in 3-D Terrain. In: Natarajan, R., Ojo, A. (eds.) ICDCIT 2011. LNCS, vol. 6536, pp. 313–324. Springer, Heidelberg (2011)

Extended Trail Reinforcement Strategies for Ant Colony Optimization

Nikola Ivkovic[1], Mirko Malekovic[1], and Marin Golub[2]

[1] Faculty of Organization and Informatics, University of Zagreb
{nikola.ivkovic,mirko.malekovic}@foi.hr
[2] Faculty of Electrical Engineering and Computing, University of Zagreb
marin.golub@fer.hr

Abstract. Ant colony optimization (ACO) is a metaheuristic inspired by the foraging behavior of biological ants that was successfully applied for solving computationally hard problems. The fundamental idea that drives the ACO is the usage of pheromone trails for accumulating experience about the problem that is been solved. The best performing ACO algorithms typically use one, in some sense "the best", solution to reinforce trail components. Two main trail reinforcement strategies are used in ACO algorithms: iteration best and global best strategy. This paper extends the reinforcement strategies by using the information from an arbitrary number of previous iterations of the algorithm. The influence of proposed strategies on algorithmic behavior is analyzed on different classes of optimization problems. The conducted experiments showed that using the proposed strategies can improve the algorithm's performance. To compare the strategies we use the Mann–Whitney and Kruskal – Wallis statistical tests.

Keywords: reinforcement strategy, pheromone trail, MAX-MIN ant system, Ant colony optimization, Swarm intelligence, combinatorial optimization, parameter settings.

1 Introduction

Ant colony optimization (ACO) [1], [2] is a class of algorithms inspired by a foraging behavior of biological ants. The colony of ants searches a surrounding area for food sources. The ants that found a food leave a pheromone trail on its way back. This way, the ants communicate indirectly with the rest of the colony by modifying a surrounding environment. The other ants then follow the pheromone trails to a food source, and leave its own trails on the way back to the colony. The shorter paths are reinforced more often, and this attracts more ants, causing an autocatalytic effect. After some time, the most of ants use the shortest path. The pheromone trails laid on the longer paths eventually evaporates.

The ACO algorithms use artificial ants to construct solutions using solution components. The solution components are linked with artificial pheromone trails that affect the solution construction process. The trails encompass the collective

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 662–669, 2011.
© Springer-Verlag Berlin Heidelberg 2011

knowledge, based on the experience of the colony about the problem. At the end of iteration the components that constitute good solutions are reinforced.

The ACO metaheuristic is successfully applied on a variety of hard computational problems. The first ant based algorithm, the Ant system (AS) [3], uses all solutions constructed in the previous iteration to reinforce the trails, but the trails associated with better solutions are reinforced more than lower quality solutions. After the AS, a number of ACO algorithms with improved performance on different combinatorial problems were published, and the most of them use only one solution, in some sense the best one, for trails reinforcement. The Ant colony system (ACS) [4] uses relatively greedy strategy and reinforces the trails of the global best solution (also known as best-so-far), i.e. the best solution that was constructed from the beginning of the algorithm. One of the most successful and popular [5] ACO algorithm, MAX-MIN ant system (MMAS) [6] usually uses the best solution constructed in the current iteration, but for bigger problems it is recommended to use the global best strategy. Considering that standard strategies are diametrically opposite, in this paper we propose new strategies that allow a finer control between explorativity of the iteration best and greediness of the global best strategy. The proposed strategies are not concerned with the pheromone values to be added on the trails and are also applicable when the amount of additional pheromone changes as the algorithm progresses [7].

This paper is organized as following. Section 2 briefly explains the ACO algorithm. In the section 3, the new strategies for selecting a solution for trails reinforcement procedure are introduced and compared. The section 4 briefly presents optimization problems used in the experimental evaluations of the new strategies and explains conditions under which the experiments were conducted. The section 5 presents and analyzes the results of the experimental researches, and section 6 gives final conclusions.

2 Ant Colony Optimization and MAX-MIN Ant System

Ant colony optimization can be described with the preudocode written in the table 1. In the `Initialize()` procedure the MMAS sets all trails to a maximum value and also the parameters of the algorithm are set. After that, the algorithm runs iteratively until satisfactory good solution is found or predefined time or number of iterations elapses. Solutions are constructed in the `ConstructSolutions()` procedure by adding solution components in the list of components until entire solution is constructed. The probability of selecting *i-th* component $c(i)$ from the set of components L_i is given for MMAS by expression (1). Parameters α and β balance between pheromone trail $\tau_c(i)$ and heuristic value $\eta_c(i)$. Update procedure includes trails evaporation using (2) for all trails, and trails reinforcement (3) for components included in the iteration best or global best solution. In the MMAS trails are maintained within a minimum and a maximum limits. The parameter ρ is a trail evaporation rate, and the $f(S^{best})$ gives a goodness of the solution used in the trails reinforcement procedure.

Table 1. The ACO algorithm and the formulas used in the MMAS

The ACO preudocode	Formulas used in the MMAS

```
Initialize()
DO until stop conditions are met
  ConstructSolutions()
  UpdateTrails()
NEXT ITERATION.
```

$$p_{c(i)} = \frac{\tau_{c(i)}^{\alpha} \cdot \eta_{c(i)}^{\beta}}{\sum_{k \in L_i} \tau_k^{\alpha} \cdot \eta_k^{\beta}} \qquad (1)$$

$$\tau_{c(j)} = (1 - \rho) \cdot \tau_{c(j)} \qquad (2)$$

$$\tau_{c(k)} = \tau_{c(k)} + \frac{1}{f(s^{best})} \qquad (3)$$

3 Extended Reinforcement Strategies

The first extended strategy, the κ-best strategy, is defined for any natural number κ. The best solution constructed in the previous κ iterations is used for reinforcement of the trails. Formally, the κ-best solution for iteration i can be defined (for minimization problems) with expression (4).

$$KappaBestSolution_i = \min_{0 \le k \le \min\{i,\kappa\}} \{IterationBestSolution_{i-k}\}\,. \qquad (4)$$

Before the algorithm reaches the iteration κ, the best solution of all constructed solutions is used for the reinforcement. For implementation of this strategy it is necessary to save up to κ solutions from the previous iterations. In the every iteration, it is necessary to save one new solution, and to delete the oldest saved solution. Therefore, it is convenient to use queue – like structure, which also allows reading all solution in the structure, when searching for the best solution.

The second strategy, the κ-best strategy, is a kind of approximation of the κ-best strategy that uses solutions from at most κ previous iterations. Initially, the best solution from the first iteration is set as a κ-max-best solution. In an every following iteration, the iteration best solution is saved as the κ-max-best solution if it is better than the previously saved one or if the κ-max-best solution has not been updated for previous κ iterations. The method of selecting a κ-max-best solution is described with the preudocode:

```
counter = counter + 1
IF ib "is better than" kappaMaxBest OR counter >= kappa DO
      counter = 0
      kappaMaxBest = ib
END IF
```

Initially, the `counter` variable is set to 0 and the best solution constructed in the first iteration is saved in the `kappaMaxBest` variable. The iteration best solution is saved in the `ib` variable.

A time complexity for the κ-best strategy is $O(\kappa)$, since it is necessary to search the list of up to κ elements, and κ-max-best strategy has $O(1)$ complexity. It can be shown that κ-best and κ-max-best strategies are generalization of standard strategies. For κ-best strategy, if the κ is equal or greater than the maximum iteration (*MAXITER*), then the solutions from all iterations are considered when searching for the best one, which is equivalent to the global best strategy. Also, for the κ-max-best strategy `counter` cannot exceed the *MAXITER*, so only a global best solution can be stored in the `kappaMaxBest` variable, if kappa>=*MAXITER*. Also, at the beginning of the algorithm, when kappa<= current iteration, all κ-best and κ-max-best exhibit an equivalent behavior to the *gb* strategy. The table 2 summarizes general equivalences between different strategies.

Table 2. The special cases equivalences between strategies

Standard strategy	κ-best strategy	κ-max-best strategy
iteration best	1-best	1-max-best
global best	∞-best	∞-max-best

4 Experimental Settings

The experimental evaluations of proposed strategies were conducted on Quadratic Assignment Problem (QAP), Travelling Salesman Problem (TSP) and Asymmetric Travelling Salesman Problem (ATSP), all well known NP-hard optimization problems that arise in many practical applications. The TSP has a set of cities (nodes) and all distances (edges) between cities are known. The problem is to find a closed tour with a minimum total distance. If the edges have directions the problem is called asymmetrical, otherwise the problem is called symmetrical. Is general TSP term should include asymmetrical and symmetrical variants, but more often TSP denotes symmetrical variant. For QAP there is a set of facilities and an equally sized set of locations with defined flow weights between the facilities and distances between the locations. The problem is to allocate the facilities to the locations in a way that the sum of products of flow weights and distances are minimized. In this paper all ATSP and TSP problems used in the empirical studies are from the *TSPLIB* library, publicly accessible at *http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/* and *VLSI Data Set* accessible at *http://www.tsp.gatech.edu/vlsi/*.

All used QAP problems are from the *QAPLIB* library publicly accessible at *http://www.seas.upenn.edu/qaplib/*.

The experiments were conducted on 18 ATSP, 19 TSP and 55 QAP problems. For every problem instance, 16 different reinforcement strategies were tested, and the experiment was repeated 100 times. Algorithm was executed with parameters: ρ=1, ρ=0.02, and number of ants was set equal to a problem size. The parameter β was set to 4 for TSPs and ATSPs, and for QAPs it was set to 0. For ATSP and TSP problems with size less than 100, the algorithm was allowed to perform the maximal 1000 iterations. For greater sized problems and all QAP problems, 10000 iterations were allowed. The experiments were performed in parallel on different processors inside *Beowulf* type cluster.

5 Results and Discussion

In the field of computational intelligence it is common to use parametric statistics, which relay on the assumption that data came from some known distribution although these assumptions are often violated [8]. By examining distributions for particular experiments we found that in some cases the distributions are very different from the standard distribution. Consequently, we use nonparametric statistics for data analysis and median instead of mean as a measure of centrality. One convenient characteristics of median is that at least 50% of samples are less than or equal to the median. It can be expected that the algorithm will find a solution with at most median value with at least 50% probability if the algorithm is executed once or at least with 75% probability if it is executed twice, etc.

By examining the functional dependencies of a median solution in the particular iterations for all tested problems (92 problems) we noticed following. The choice of the optimal strategy depends on a problem that is being solved an also on the maximal allowed number of iterations. The typical examples are presented on Fig. 1 (a)-(c).

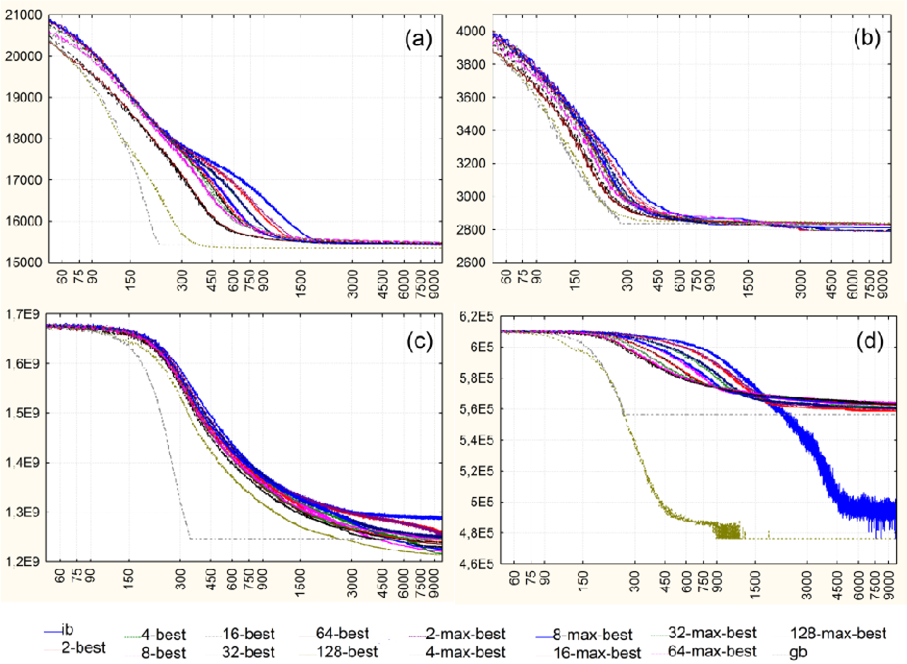


Fig. 1. Median iteration best solution on the axis ordinate, as a function of iteration, for different strategies for: rd400 TSP problem (a), ftv17 ATSP problem (b), and QAP problems tai100b (c) and lipa40b (d). The scale on the abscissa axis is logarithmic.

At the beginning, all strategies cause similar algorithmic behavior because differences in the pheromone trails are still small. Also, some strategies have equivalent behavior on the beginning. After that, performance of the strategies differs more considerably. Greedier strategies typically find good solutions faster than less greedy strategies. For problems that are relatively easy for MMAS algorithm, greedier strategies often find an optimal solution relatively fast and less greedy strategies catch with them later. In that case, greedier strategies are the optimal choice regardless of the iteration. For harder problems, more often less greedy strategies outrun the greedier strategies after some number of iterations, so the choice of the optimal strategy depends on the number of iterations.

For rd400 problem (Fig. 1a), the *gb* strategy gives the best results for roughly 150-350 iterations and then it is outrun with 128-best strategy that gives the best solutions from 350-10000 iterations. For ftv170 (Fig. 1b) at first, *gb* and 128-best give the best results, but at the end, the best results are achieved by *ib*, 2-best, 4-best and 2-max-best strategies, while the most greedy strategies achieves the worst results. The *gb* strategy greatly outstands as the best strategy for tai100b (Fig 1c) at the beginning of the algorithm. This is typical for QAP problems and much more noticeable than with TSPs and ATSPs, possibly because the MMAS does not use the heuristic values for QAPs. After about 2500 iterations 128-best strategy outrun the *gb* strategy and stay the best one until the end of 100000 iterations. At the end, 4-best strategy comes close to 128-best as the second best strategy.

In the case when the difference between strategies is the most noticeable, the bigger κ parameter means more successful κ-best strategy: The same applies for the κ-max-best strategy. For lower κ values, the performance of κ-best and κ-max-best strategies is similar, but for bigger κ, i.e. 64 and 128, the κ-best is noticeably better than the κ-max-best.

As an exception from other tested problems, for lipa40b (Fig 1d) the MMAS algorithm shows rather unexpected behavior. At first, the *gb* and 128-best achieves the best solutions, but after about 300 iterations the *gb* strategy stagnates and 128-best keeps finding better solutions relatively fast. After while, *ib* strategy starts finding better solutions relatively fast, and at the end, the both 128-best and *ib* strategies have median solutions equal to the optimal one, while the other strategies achieve considerably poorer solutions.

To compare κ-best and κ-max-best strategies for all tested problems at ones, we divided values of the achieved solutions with the values of the best known solutions for particular problems and run Mann–Whitney U test. The test was run for κ-best and κ-max-best pairs after 300, 1000, 3000 and 10000 iterations (for TSP and ATSP only for 300 and 1000 iterations, since for problems with size up to 100 maximum number of iteration was limited to 1000). For QAP problems, the Mann–Whitney test confirmed with very high significance ($p<0.000001$) that 128-best have a better performance than the 128-max-best strategy. Also, it was confirmed with the high significance ($p<0.01$) that 64-best is a better than 64-max-best. For other κ values, the test could not confirm that κ-best and κ-max-best exhibit the significant difference in the performance ($p>0.5$ and is often rather close to 1). The test confirmed with the high significance ($p<0.01$) that the κ-best is better than the κ-max-best for the ATSP for all κ values (except 1 and ∞), and for the TSP for κ values greater than 8.

To compare the performance of different strategies we use Kruskal-Wallis *H* test. The strategies with highest rank score are presented with scatterplots on Fig. 2.

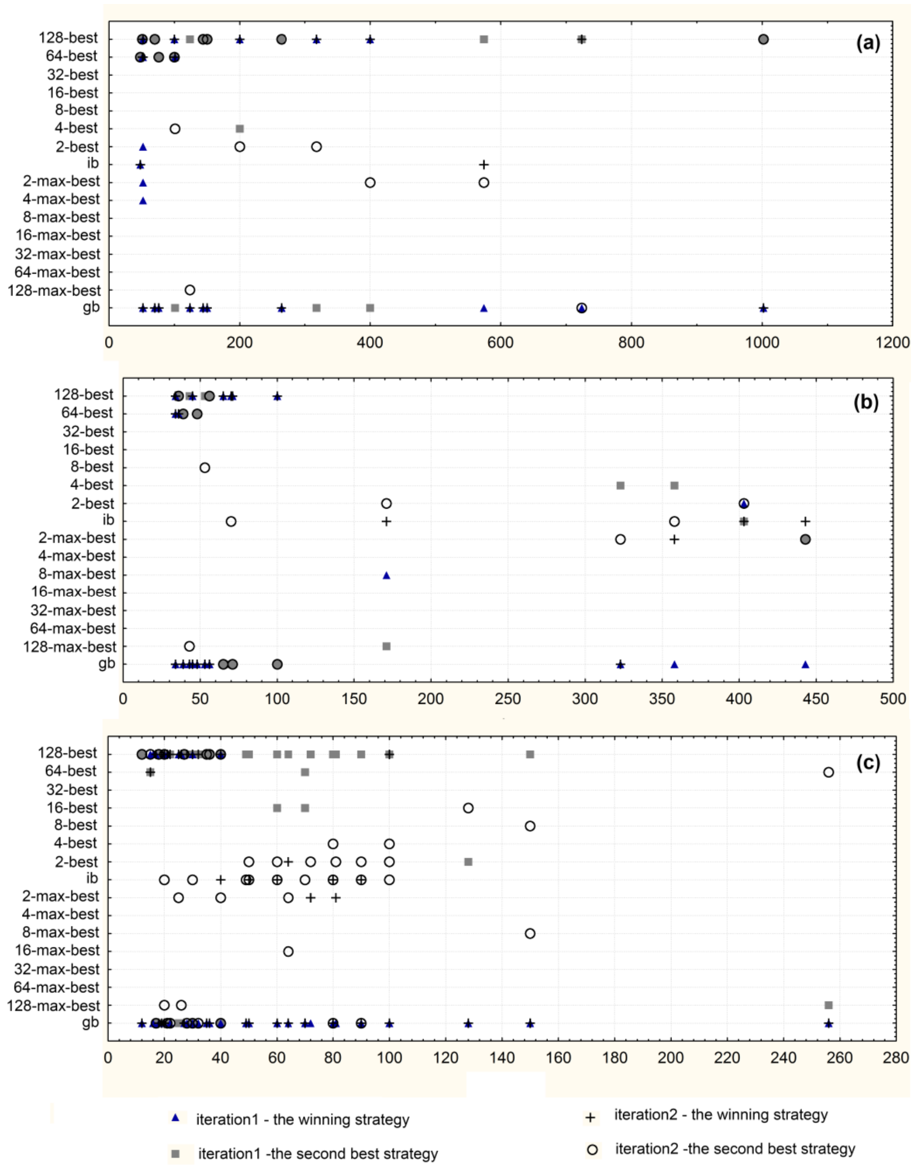


Fig. 2. The winning and the second best strategies as a result of Kruskal-Wallis statistical test for TSP (a), ATSP (b) and QAP (c). The problem size is denoted on the axis abscise.

For few easiest problems (e.g. ftv17), the algorithm found the optimal solutions very fast with all strategies. In these cases, the Kruskal-Wallis test had low H value and high p value. Consequently, these problems were excluded from the scatterplots. The *iteration1* for ATSP and TSP is equal to 300 iterations for problems with size up to 100, and 1000 iterations for bigger problems. The *iteration2* is set to 1000iterations for smaller problems and 10000 for bigger problems. This is motivated by the fact that for smaller problems the algorithm coverage faster. For the QAP problems, the size of the problem seems less related to problem's hardness and a speed of convergence, so for all problems *iteration1* and *iteration2* are set to 1000 and 10000 iterations, respectfully. The results clearly show that in many cases the extended strategies are better than the standard strategies. For smaller sized problems, 128-best and *gb* strategies often give the best results. For bigger problems it seems that 128-best and *gb* give the best results at the beginning, but if algorithm is allowed to run for a longer time, strategies with lower κ outperform the greedier strategies. The middle value κ gives moderate results regardless of the iteration, but are also rarely the worst strategies.

6 Conclusion

The proposed new strategies provide a fine control of the greediness through the κ parameter. The paper analyzes an influence of the strategies on the algorithmic behavior. The statistical tests conducted on experimental data confirmed with a high significance that the performance of the ACO algorithm can be enhanced by proposed strategies. For bigger κ values the κ-best strategy has a better performance than the κ-max-best strategy, but for lower κ values the difference in not significant.

References

1. Dorigo, M., Di Caro, G.: Ant colony optimization: A new meta-heuristic. In: Angeline, P.J., et al. (eds.) Proceedings of the Congress on Evolutionary Computation, vol. 2, pp. 1470–1477. IEEE Press (1999)
2. Dorigo, M., Di Caro, G., Gambardella, L.M.: Ant Algorithms for Discrete Optimization. Artificial Life 5(2), 137–172 (1999)
3. Dorigo, M., Maniezzo, V., Colorni, A.: The Ant System: Optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics – Part B 26, 29–42 (1996)
4. Dorigo, M., Gambardella, L.M.: Ant Colony System: A Cooperative Learning Approach to the Travelling Salesman Problem. IEEE Transactions on Evolutionary Computation 1(1), 53–66 (1997)
5. Dorigo, M.: Ant colony optimization. Scholarpedia 2(3), 1461, revision #82084 (2007)
6. Stützle, T., Hoos, H.H.: MAX–MIN Ant System. Future Generation Computer Systems 16(9), 889–914 (2000)
7. Abraham, A., Konar, A., Samal, N.R., Das, S.: Stability analysis of the ant system dynamics with non-uniform pheromone deposition rules. IEEE Congress on Evolutionary Computation, 1103–1108 (2007)
8. Derrac, J., García, S., Molina, D., Herrera, F.: A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm and Evolutionary Computation 1(1), 3–18 (2011)

Fractional-Order $PI^{\lambda}D^{\mu}$ Controller Design Using a Modified Artificial Bee Colony Algorithm

Anguluri Rajasekhar[1], Vedurupaka Chaitanya[1], and Swagatam Das[2]

[1] Department of Electrical and Electronics Engg, NIT, Warangal, India
[2] Electronics and Communication Sciences Unit, Indian Statistical Institute, Kolkata, India
`rajasekhar.anguluri@ieee.org`, `chaitu266@gmail.com`,
`swagatam.das@ieee.org`

Abstract. Artificial Bee Colony (ABC) algorithm is one of the most recent swarm intelligent foraging based algorithms for real parameter optimization over continuous spaces. It has shown great consistency in outperforming different types of evolutionary approaches when tested over various instances, from traditional benchmark functions to real-world practical problems. This paper describes the design of *Fractional-Order Proportional-Integral-Derivative (FOPI)* controllers, using a variant of ABC, known as C-ABC (Cauchy Mutated Artificial Bee Colony Algorithm). FOPID controllers' parameters are composed of the proportionality constant, integral constant, derivative constant, derivative order and integral order, and its design is more complex than that of conventional integer-order proportional-integral-derivative (PID) controller. In this approach controller synthesis is based on user specifications like peak overshoot and rise time; which are used to develop a single objective optimization problem. Simulation results for some real world practical instances and, comparison of the same for C-ABC and few established optimization techniques (Basic ABC, Particle Swarm Optimization (PSO) and Genetic Algorithm (GA)) have been demonstrated to show the superiority of the proposed design technique.

Keywords: Artificial Bee colony Algorithm, Fractional-order PID controller, Cauchy Mutations, Particle Swarm Optimization, Genetic Algorithm.

1 Introduction

Dynamic systems based on fractional order calculus [1,2] have been a subject of extensive research in recent years [3,4], since the proposition of concept of the fractional-order controllers and then demonstration of their effectiveness in actuating desired fractional order system responses by Podlubny in 1999 [5]. A few recent works in this direction as well as schemes for digital and hardware realizations of such systems can be figured in Chen *et al* [6, 7], and Nakgawa and Sorimachi [8]. Petras [9] came up with a method based on the pole distribution of the characteristic equation in the complex plane. The fractional controller can also be synthesized by cascading a proper fractional unit to an integer-order controller (Chengbin and Hori, [10]).

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 670–678, 2011.
© Springer-Verlag Berlin Heidelberg 2011

Proportional-integral-derivative (PID) controllers have been used for several decades in industries for process control applications. The reason for their wide popularity lies in the simplicity of design and good performance including low percentage overshoot and small settling time for slow process plants [11]. In *fractional-order proportional-integral derivative (FOPID) controller,* I and D operations are usually of fractional order; therefore, besides setting the proportional, derivative and integral constants $K_{P,}$ K_d ,K_I ,we have two more parameters: the order of fractional integration μ and that of fractional derivative λ. Determining an optimal set of values for K_P , K_d , K_I ,μ and λ to meet the user specification for a given process plant calls for real parameter optimization in five-dimensional hyperspace.

Artificial Bee Colony (ABC) algorithm is one of the most recent, swarm intelligence based algorithms which simulates the foraging behavior of honey bee colonies, for solving dynamic numerical optimization problems. ABC was suggested by Karaboga and Basturk for function optimization in 2005 [12]. In this paper we propose a modified ABC algorithm in which the trapped bees near local attractor basin are forced to some nearby area, to locate other food sources (potential solutions) by applying a small perturbation. This perturbation is applied using Cauchy distribution and hence forth this modification is expected to help ABC bees in escaping the local attractor and also in improving the convergence rate. We termed this variant as C-ABC (ABC with Cauchy Mutation)

The optimization-based design process has been tested for actuating the response of four process plants which are of integer order. The performance of the C-ABC based $PI^{\lambda}D^{\mu}$ controller has been compared with other fractional-order controllers designed with that of ABC, PSO and GA [13, 14]. This comparison delineates the superiority of the proposed method in terms of quality of final solution, convergence speed and robustness.

2 Fractional Order Systems

Fractional Calculus (FC) is the branch of mathematics having 300 years of history and which is gained the practical importance in recent years. FC is a generalization of ordinary differential calculus which considers the possibility of taking real number power of differential and integration operator. Generalized form of differ-integrator may be given as

$$ {}_aD_t^q f(t) = \frac{d^q f(t)}{[d(t-a)]^q} \tag{1} $$

Where q represents the real order of the differintegral, t is the parameter for which the differintegral is taken and a is the lower limit, in general considered to be 0 in almost all cases. Amongst all the definitions put forward for diiferintegral, Caputo's [15] definition for fractional derivative of order λ with respect to variable t is found to be most suitable in applications of fractional order controller design. The definition is as follows as

$$ {}_0D_t^{\lambda} y(t) = \frac{1}{\Gamma(1-\delta)} \int_0^t \frac{y^{(m+1)}(\tau)d\tau}{(t-\tau)^{\delta}} \qquad (\gamma = m+\delta, m \in Z, 0 < \delta \le 1) \tag{2} $$

where $\Gamma(Z)$ is Euler's gamma function. For $\gamma < 0$, the formula for fractional integral of order $-\gamma$ is given as

$$_0D_t^{-\gamma} y(t) = \frac{1}{\Gamma(-\gamma)} \int_0^t \frac{y(\tau)d(\tau)}{(t-\tau)^{1+\gamma}} \quad (\gamma < 0) \tag{3}$$

Using Caputo's definition a fractional-order system may be defined by the following n-terms inhomogeneous fractional differential equation (FDE)

$$a_n D^{\beta_n} y(t) + a_{n-1} D^{\beta_{n-1}} y(t) + ... + a_0 D^{B_0} y(t) = u(t) \tag{4}$$

Frequency domain transfer function of the system may be obtained through Laplace Transform of fractional derivatives as

$$G_n(s) = \frac{1}{a_n s^{\beta_n} + a_{n-1} s^{\beta_{n-1}} + ... a_0 s^{\beta 0}} \tag{5}$$

3 Artificial Bee Colony Algorithm

Artificial bee colony algorithm classifies the foraging artificial bees into three groups; the *employed bees, the onlooker bees* and the *scouts.* A bee that is currently searching for food or exploiting a food source is called an employed bee and a bee waiting in the hive for making decision to choose a food source is called an onlooker. For every food source, there is only one employed bee and employed bee of abandoned food source becomes scout.

The algorithm starts by initializing all *employed bees* with randomly generated food sources (solutions). For each iteration, employed bee finds a food source in the neighborhood of its current food source and evaluates its nectar amount i.e., (*fitness*). In general the position of i_{th} food source is represented as $X_i = (x_{i1}, x_{i2}, ..., x_{iD})$. After the information is shared by the employed bees, *onlooker bees* go to the region of food source at X_i based on the probability p_i defined as

$$P_i = \frac{fit_i}{\sum_{k=1}^{FS} fit_k}, \tag{6}$$

where FS is total number of food sources. Fitness value fit_i is calculated by using following equation.

$$fit_i = \frac{1}{1 + f(X_i)}. \tag{7}$$

Here $f(X_i)$ is the objective function considered. The onlooker finds its food source in the region of X_i by using following equation.

$$x_{new} = x_{ij} + r * (x_{ij} - x_{kj}). \tag{8}$$

If the obtained new fitness value is better than the fitness value achieved so far, than the bee moves to this new food source leaving this old one otherwise it retains its old food source. When all employed bees have completed this process, the information is shared with onlookers. Each of the onlookers selects food source according to probability given above. By this scheme good sources are well accommodated with onlookers. Each bee will search for a better food source for a certain number of cycles (*limit)*, and if the fitness value doesn't improve then that particular bee becomes scout bee. The food source is initialized to that *scout bee* randomly.

4 Modified Artificial Bee Colony Algorithm

The focus of this paper is to introduce a mechanism which will not only keep a track of the progress of bees but will also help the individuals in escaping the local attractor basin by allowing them to scatter to a new region. In order to keep a record of the success of bees, in the present study, we introduce a concept of *bee-scanner*. The work of this scanner is to identify the bees which are unable to locate potential food sources for successive generations/iterations. If there is no improvement in fitness, then *bee_scanner* is increased by unity in each generation/iteration. This process is repeated until we achieve user-defined value of highest scan (HS). Once HS is attained, it indicates that the position of the bees should be changed. To do this Cauchy movements are applied [16] for which the probability density function (PDF) is given by the following equation:

$$f(x; x_0; \gamma) = \frac{1}{\pi\gamma\left[1 + \left(\frac{x - x_0}{\gamma}\right)^2\right]} = \frac{1}{\pi}\left[\frac{\gamma}{(x - x_0)^2 + \gamma^2}\right] \tag{9}$$

where x_0 is the location parameter, specifying the location of the peak of the distribution, and γ is the scale parameter which specifies the half-width at half-maximum. For large values of γ we get a fat tail curve, whereas for smaller values of γ, the shape of the curve changes towards a sharper peak. In this study we have taken γ as 0.1, which will produce a very sharp peak, resulting in a small area around the search space which will help the employed bee and onlooker bees in exploiting the region near the available food sources.

A new food source $X_i = \{x_{i1}, x_{i2}, \ldots, x_{iD}\}$ by C-ABC is generated as

$$x_{new} = \begin{cases} x_{ij} + r * \left(x_{ij} - x_{kj}\right) & \text{if } bee\_scanner < HS \\ x_{i,best} + C(\gamma, 0) & \text{else} \end{cases} \tag{10}$$

Where $C(\gamma, 0)$ stands for random number generated by Cauchy probability distribution with scale parameter γ and centered at origin. After generation of new food source, optimization process, similar to that of basic ABC is used. This modification allows the algorithm to maintain a better balance between the exploration and exploitation factors.

5 Design of Fractional $PI^{\lambda}D^{\mu}$ Controller Using C-ABC

A PID controller is a generic control feedback. In this process proportional control, integral control and, derivative control trio are used together to minimize the error obtained between a measured process variable and a set value. Transfer function of an integer-order PID control may be given as

$$G_c = K_P + \frac{K_I}{s} + K_d s \tag{11}$$

However, incorporating fractional orders for integral and derivative control increases the controllability of the system and thus transfer function for fractional-order PID controller may be defined as

$$G_c = K_P + K_I s^{-\lambda} + K_d s^{\mu} \tag{12}$$

Here the parameters are K_P, K_I, K_d, μ and λ. The design approach carried out in this work is based on root locus method (dominant poles method) of PID controllers. Let the closed-loop transfer function be

$$\frac{C(s)}{R(s)} = \frac{G(s)}{1 + G(s)H(s)} \tag{13}$$

where the transfer function of the process to be controlled is $G_p(s)$ and a that of the controller is $G_c(s)=U(s)/E(s)$ and $G(s)=G_c(s)G_p(s)$. We assume unity feedback gain, i.e., $H(s) = 1$. From Eq. (13) the characteristic equation of the closed-loop system is

$$1 + G_P(s)G_c(s) = 0 \tag{14}$$

Dominant poles of the system are the roots of this equation, and hence from (12) we can say that

$$1 + [K_p + K_I(-x + jy)^{-\lambda} + K_d(-x + jy)^{\mu}]G_p(-x + jy) = 0 \tag{13}$$

If I be the imaginary part and R be the real part of the expression (14) and, ψ the phase angle = $\tan^{-1}(I/R)$, the fitness function is defined as

$$J(K_P, K_I, K_d, \lambda, \mu) = |I|^2 + |R|^2 + |\psi|^2 \tag{14}$$

Our aim is to find out an optimal solution set $(K_P, K_I, K_d, \mu, \lambda)$ for which $J = 0$. To find the appropriate set of parameters that leads to optimal value of J, a five dimensional search space is selected for C-ABC; the dimensions being $(K_P, K_I, K_d, \mu, \lambda)$ and in case of C-ABC for the given five dimension the i_{th} food source is represented as it may be represented as

$$X_i = [K_p, K_I, K_d, \lambda, \mu] \tag{15}$$

Practical considerations of PID controller require that the parameter set adheres to the following constraints

$$\begin{aligned} &1 \le K_P \le 1000 \\ &0 \le \lambda, \mu \le 1 \\ &1 \le K_I \;\; K_d \le 500 \end{aligned} \tag{16}$$

6 Experimental Results and Validation

The proposed method has been tested on four specific instances of controller design for real life analog plant, which are summarized in Table 1.For fractional-order differentiator s^r is discretized using Tustin-based Continuous Fraction Expression (CFE) which may be given as

$$D^{\pm r}(z) = \left(\frac{2}{T}\right)^{\pm r} \frac{P_p(z^{-1})}{Q_q(z^{-1})} \tag{17}$$

Where, T is the sampling period and, value of the expression for different degrees (i.e., different values of p and q). In this paper we had considered sampling period to be $T=0.01$ *sec* and also we had made an approximation to be order as $p=q=5$.

Table 1. Description of problem instances considered

Problem No.	Process plant transfer function $G_p(s)$	User Specifications	
		Maximum Overshoot (%)	Rise Time (sec)
I	$\dfrac{k}{(Js+b)(Ls+R)+k^2}$ $J=0.01,\ b=0.1,\ k=0.01,\ R=1,\ L=0.5$	5	0.5
II	$\dfrac{400}{(s^2+50s)}$	10	0.1
III	$\dfrac{1553}{(s^2+14s+40.2)}$	10	0.1
IV	$\dfrac{250s+500}{(s^3+12s^2+100s)}$	15	0.1

Table 2. Designed Controller Plants Using C-ABC

Process plant transfer function $G_p(s)$	**Controller transfer function $G_c(s)$**
$\dfrac{k}{(Js+b)(Ls+R)+k^2}$ *J=0.01, b=0.1, k=0.01, R=1, L=0.5*	$9.9979+s^{-0.9077}+1.8526s^{0.4032}$
$\dfrac{400}{(s^2+50s)}$	$5.1273+1.3917s^{-0.1415}+0.7370s^{0.5103}$
$\dfrac{1553}{(s^2+14s+40.2)}$	$4.0465+0.8612s^{-0.6862}+0.9740s^{0.8217}$
$\dfrac{250s+500}{(s^3+12s^2+100s)}$	$2.9991+1.4920s^{-0.5733}+0.9169s^{0.3399}$

Table 3. Summary of the performance of closed-loop system under different PID controllers against the unit step

Plant	**Different Controllers Used**	**Unit Step Response**			**Fitness Value**
		Max Overshoot (%)	Rise Time (sec)	Steady State error	
I	FOPID controller using C-ABC	**1.9824**	0.8532	**0.4312**	0.005247
	FOPID controller using PSO	3.91	0.822	1.9	0.0001
	FOPID controller using GA	6.31	**0.695**	2.1	0.0287
II	FOPID controller using C-ABC	3.6244	**0.0283**	**3.346e-06**	2.47186e-05
	FOPID controller using PSO	**3.109**	0.89	1.92	0.0001
	FOPID controller using GA	5.07	0.564	2.2	0.0312
III	FOPID controller using C-ABC	**0.5584**	**0.0597**	**0.004445**	0.001049
	FOPID controller using PSO	4.17	0.48	1.45	0.0001
	FOPID controller using GA	5.94	0.36	1.9	0.0480
IV	FOPID controller using C-ABC	**0.212**	**0.0291**	**8.211e-05**	0.007807
	FOPID controller using PSO	3.67	0.45	1.56	2.880545
	FOPID controller using GA	4.95	0.39	1.89	0.0145

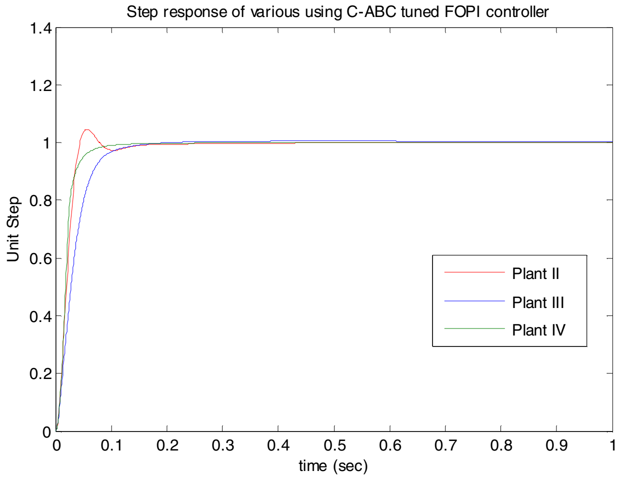


Fig. 1. Unit Step response of closed loop systems for the test problems II III IV

7 Conclusion

In the present study a new variant of ABC algorithm is proposed for designing fractional order PID controllers. The Superiority of proposed algorithm is demonstrated by comparing it with other state of art optimization techniques like PSO and GA over performance indices like peak overshoot, settling time etc., Our future research will focus on applying ABC (and also its variants) in designing the fractional order plants.

References

1. Oldham, K.B., Spanier, J.: The Fractional Calculus: Integrations and Differentiations of Arbitrary Order. Academic Press, New York (1974)
2. Lubich, C.H.: Discretized fractional calculus. SIAM J. Math. Anal. 17(3), 704–719 (1986)
3. Oustaloup, A.: Fractional order sinusoidal oscillators: Optimization and their use in highly linear FM modulation. IEEE Transactions on Circuits and Systems 28(10), 1007–1009
4. Chengbin, M., Hori, Y.: The application of fractional order PID controller for robust two-inertia speed control. In: Proceedings of the 4th International Power Electronics and Motion Control Conference, Xi'an (August 2004)
5. Podlubny, I.: Fractional-order systems and PI/sup /spl lambda//D/sup /mu//-controllers. IEEE Transactions on Automatic Control 44(1), 208–214 (1999)
6. Chen, Y.Q., Xue, D., Dou, H.: Fractional calculus and biomimetic control. In: Proceedings of the First IEEE International Conference on Robotics and Biomimetics (RoBio 2004). IEEE, Shengyang (2004)
7. Chen, Y.Q., Ahn, H., Xue, D.: Robust controllability of interval fractional order linear time invariant systems. In: Proceedings of the ASME 2005 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Paper # DETC2005-84744, Long Beach, CA, September 24-28, pp. 1–9 (2005)

8. Nakagawa, M., Sorimachi, K.: Basic characteristics of a fractance device. IEICE Transactions on Fundamentals E75-A (12), 1814–1819 (1992)
9. Petras, I.: The fractional order controllers: methods for their synthesis and application. Journal of electrical Engineering 50(9-10), 284–288 (1999)
10. Chengbin, M., Hori, Y.: The application of fractional order PID controller for robust two-inertial speed control. In: Proceedings of the 4th International Power Electronics and Motion Control Conference (2004)
11. Astrom, K., Hagglund, T.: PID controllers: Theory, Design and Tuning. Instrument Society of America, Research Triangle Park (1995)
12. Karaboga, D., Basturk, B.: A powerful and efficient Algorithm for Numerical Optimization: Artificial Bee Colony (ABC) Algorithm. Journal of Global Optimization 39(3) (2007)
13. Cao, J., Liang, J., Cao, B.: Optimization of fractional order PID controllers based on genetic algorithms. In: Proceedings of the International Conference on Machine Learning and Cybernatics, Guangzhou, pp. 18–21 (August 2005)
14. Holland, J.H.: Adaption in Natural and Artificial Systems. University of Michigan Press, Ann Harbor (1975)
15. Caputo, M.: Elasticita e Dissipacione, Zanichelli, Bologna (1969)
16. Stacey, A., Janic, M., Grundy, I.: Particle swarm optimization with mutation. In: Proceedings of IEEE Congress on Evolutionary Computation, pp. 1245–1439 (2003)

Reconfiguration of Distribution Systems for Loss Reduction Using the Harmony Search Algorithm

A.Y. Abdelaziz[1], Reham A. Osama[1], S.M. El-Khodary[1], and Bijaya Ketan Panigrahi[2]

[1] Department of Electrical Power & Machines, Faculty of Engineering,
Ain Shams University, Cairo, Egypt
[2] Department of Electrical Engineering, Indian Institute of Technology, Delhi, India

Abstract. This paper introduces the Harmony Search (HS) algorithm to solve the distribution network minimum loss reconfiguration problem. The HS is an emerging music-based meta-heuristic optimization algorithm. It is a population-based approach which mimics musicians' behaviors such as random play, memory-based play, and pitch-adjusted play when they perform improvisation. The HS algorithm is simple in concept, less in parameters, and easy in implementation. The optimization problem is formulated taking into account the operational constraints of the distribution systems. Results of numerical tests carried out on a 69-bus system from literature are presented to show the effectiveness of the proposed approach.

1 Introduction

Electric utilities are constantly looking for new technologies that enhance power delivery performance. One of several important issues is the control of power loss. Network reconfiguration is thought to be an effective method that can make use of existing equipment to reduce distribution feeder losses and improve system security. The reconfiguration of distribution feeders is the process of changing the system topology by opening and closing of switches. Two types of switches are used in primary distribution systems. There are normally closed switches "sectionalizing switches" and normally opened switches "tie switches". 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 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, 5], Tabu Search [6], etc. A survey of the early state of art is provided in [7]. In this paper, a recently developed musicians` behavior-inspired evolutionary algorithm, Harmony Search (HS), is adopted. It is inspired by the observation that the aim of music is to search for a perfect state of harmony which is analogous to find the optimality in an optimization process [8, 9]. This paper is organized as follows; Section (2) explains the distribution system minimum loss reconfiguration problem, its objective function and constraints; Section (3) illustrates the main steps of the HS algorithm, its

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 679–687, 2011.
© Springer-Verlag Berlin Heidelberg 2011

operators and parameters and how it is formulated to solve the network reconfiguration problem for loss reduction; Section (4) shows the numerical results of applying the HS algorithm to two test systems from literature. Finally, the conclusion is given in Section (5).

2 Formulation of the Network Reconfiguration Problem for Loss Reduction

The reconfiguration problem can be formulated as follows:

$$\text{Min } f = \sum_{i=1}^{N_R} R_i |I_i|^2 \tag{1}$$

Subject to the following constraints:

- The voltage magnitude

$$V_{\min} \leq |Vi| \leq V_{\max} \; ; \forall \, i \in N_b \tag{2}$$

- The current limit of branches

$$|Ii| \leq I_{j \max} \; ; \forall \, j \in N_R \tag{3}$$

- 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_j and I_{jmax} 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 [10]. This method has excellent convergence characteristics and is very robust and proved to be efficient for solving radial distribution networks.

3 Harmony Search Algorithm

The HS algorithm is a new meta-heuristic population search algorithm proposed by Geem et al. [11, 12]. HSA was derived from the natural phenomena of musicians' behavior when they (population members) collectively play their musical instruments (decision variables) to come up with a pleasing harmony (global optimal solution). This state is determined by an aesthetic standard (fitness function). When a musician is improvising, he has three possible choices; playing any famous tune exactly from his memory (Memory Consideration); playing something similar to the aforementioned tune (pitch adjustment); composing new or random notes from the pitch range (Random Selection). The main steps of HS are as follows:

- Step (1): Initialize the problem, algorithm parameters and harmony memory

The optimization problem is specified as follows:

$$\text{Minimize } f(x), \text{subjected to } x_i \in X_i, i = 1,2,3,....,N \tag{4}$$

where f(x) is an objective function, x is the set of each decision variable xi , N is the number of decision variables, Xi is the set of the possible range of values for each decision variable. The HS algorithm parameters are also specified in this step. These are the harmony memory size (HMS); Harmony Memory Considering rate (HMCR); Pitch Adjusting Rate (PAR) and the maximum number of improvisations (NI). The harmony memory (HM) is a memory location where all the solution vectors (sets of decision variables) are stored. Here, HMCR and PAR are parameters that are used to improve the solution vector, which are defined in Step 2. The initial HM consists of a certain number of randomly generated solutions for the optimization problem under consideration without violating the constraints. For a problem of *N* variables, a HM with the size of HMS can be represented as follows:

$$HM = \begin{bmatrix} x1^{1} & x2^{1} & .... & xN-1^{1} & xN^{1} & : & f1 \\ x1^{2} & x2^{2} & .... & xN-1^{2} & xN^{2} & : & f2 \\ .. & .. & .... & .. & .. & : & .. \\ x1^{HMS} & x2^{HMS} & & xN-1^{HMS} & xN^{HMS} & : & fHMS \end{bmatrix} \tag{5}$$

where $(x_1^i, x_2^i ... x_N^i)$ represents a candidate solution for the optimization problem and f_1 is the value of the fitness function corresponding to the first solution vector. For the network reconfiguration problem, the solution vector is represented by the set of tie switches in the radial configuration of the network [13]. The configuration of the 69-bus system which is shown in fig. (2) can be represented as in (6).

$$Configuration = [59 \quad 62 \quad 71 \quad 70 \quad 15] \tag{6}$$

where 59 is the tie switch from loop 1, 62 is the tie switch from loop 2, etc. The HM is sorted in ascending order with respect to the fitness function such that configuration with the least power loss (best configuration) is at the top and the one with the highest power loss (worst configuration) is at the bottom.

- Step (2): Improvise a new harmony

A new harmony vector $(x_1', x_2' ... x_N')$ is generated based on three main rules: (1) memory consideration (2) pitch adjustment and (3) random selection. Generating a new harmony is called 'improvisation'. Each component of the solution is chosen either from the harmony memory or by randomness depending on the value of the HMCR, which varies between 0 and 1, and defined as the rate of choosing one value from the historical values stored in the *HM*, while (1-*HMCR*) is the rate of randomly selecting one value from the possible range of values as follows:

$$\begin{array}{l} If\,(rand\,(\,)\langle\, HMCR) \\ x_i^{`} \leftarrow x_i^{`} \in \{x_i^{1}, x_i^{2} \ldots\ldots x_i^{HMS}\} \\ else \\ x_i^{`} \leftarrow x_i^{`} \in Xi \\ end \end{array} \tag{7}$$

where *rand ( )* is a uniformly distributed random number between 0 and 1and X_i is the set of the possible range of values for each decision variable. Every variable $x_i^{'}$ obtained by the memory consideration is examined to determine whether it should be pitch-adjusted. This operation uses the PAR parameter, which is the rate of pitch adjustment and the value (1-PAR) is the rate of doing nothing as follows:

$$\begin{array}{l} If\,(rand\,(\,)\langle\, PAR) \\ x_i^{`} \leftarrow x_i^{`} \pm rand(\,)\ {*}BW \\ else \\ x_i^{`} \leftarrow x_i^{`} \\ end \end{array} \tag{8}$$

where *BW* is an arbitrary distance bandwidth for the continuous design variable and *rand ( )* is uniform distribution between -1 and 1. If the problem is discrete in nature as the network reconfiguration problem, *BW* is taken as 1.

- Step (3): Update Harmony memory

If the new harmony vector ($x_1^{'}$, $x_2^{'}$, …, $x_N^{'}$) is better than the worst harmony in the HM, judged in terms of the objective function value (yields to a better fitness than that of the worst member in the HM), the new harmony is included in the HM and the existing worst harmony is excluded from the HM. The HM is rearranged in ascending order according to the fitness function. Otherwise, the new harmony is discarded.

- Step (4): Check stopping criteria

If the stopping criterion (maximum number of improvisations) is satisfied, computation is terminated and finally the best one among the solution vectors stored in the HM is selected, which is the optimum solution of the problem. Otherwise, Steps 3 and 4 are repeated.

4 Worked Examples

A 69-bus system from literature is reconfigured using the HS algorithm and the results are compared with previously applied algorithms. The solution algorithm was implemented using MATLAB V7. The system is a 12.66 KV radial distribution system with one supply point, 69 buses, 7 laterals and 5 tie switches. The system data are given in [4]. The total substation loads of the base configuration which is shown in Fig. (1) are 1107.9 KW and 897.9 KVAR. The system was reconfigured with normal, heavy and light loading conditions. In the normal loading condition, the system`s real and reactive load demands of each node are used without any change. For the heavy and light loading conditions, the system`s real and reactive load demands of each node are multiplied by a constant equal to 1.2 and 0.5 respectively to construct heavy and light load system patterns. The HS parameters used for the three loading cases are NI=200, HMS=10, HMCR=0.8, PAR=0.2. Tables (1), (2) and (3) show the initial configuration data of the system and results obtained by reconfiguration using the proposed HS algorithm and other algorithms from literature for the three loading patterns. The initial power loss was 20.8 KW, 30.358 KW and 5.0927 KW respectively. Reconfiguration of the system using the proposed algorithm resulted in the same configuration for the three cases [10-70, 12-20, 13-14, 47-48, 50-51] which is shown in Fig. (2) with a 54.6 %, 55.01 % and 54.38 % reduction in losses respectively.

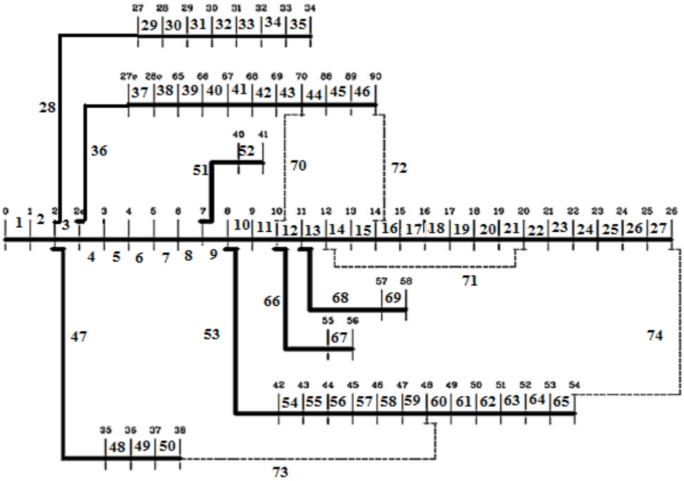


Fig. 1. Initial configuration of the 69-bus system. Tie switches and sectionalizing switches are represented by dotted and solid lines respectively.

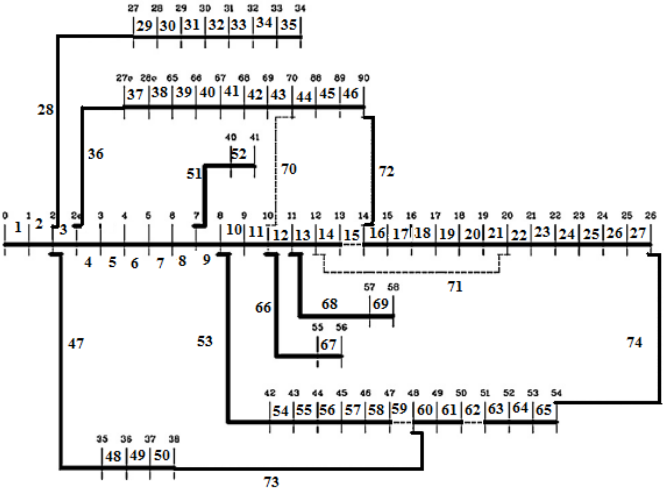


Fig. 2. Final configuration of the 69-bus system

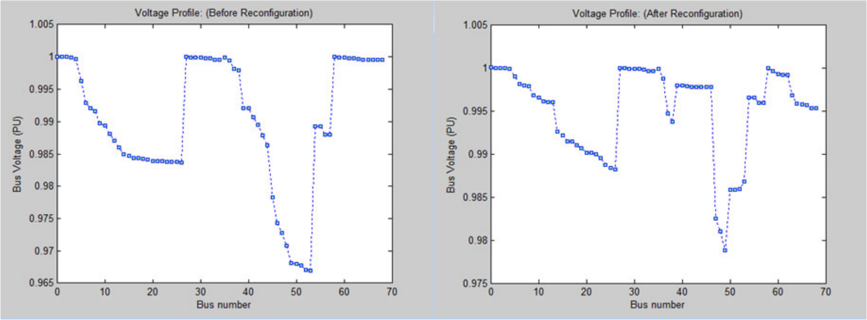


Fig. 3. Voltage profile of the 69-Bus system before and after reconfiguration (normal load)

Table 1. Reconfiguration results of the 69-bus system (normal load)

Reconfiguration	Losses (KW)	Tie Switches
Initial configuration	20.8	[10-70, 12-20, 14-90, 38-48, 26-54]
Final configuration using HS	Best 9.42 Worst 9.9 Average 9.44	[10-70, 12-20, 13-14, 47-48, 50-51]
Final configuration using SA [4]	9.42	[10-70, 12-20, 13-14, 45-46, 50-51]
Final configuration using MPS[3], SA [5] and MTS[6]	9.42	[10-70, 12-20, 13-14, 44-45, 50-51]

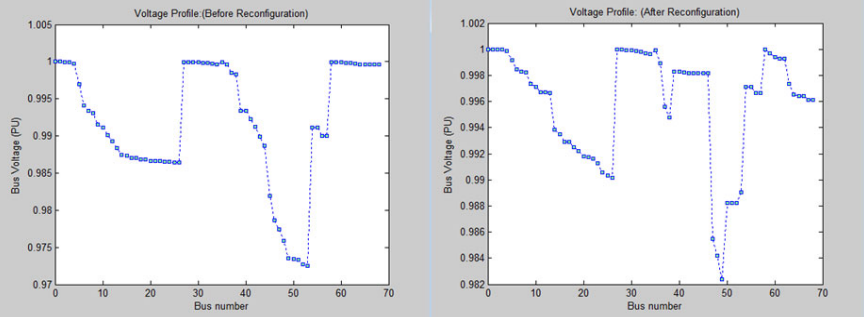


Fig. 4. Voltage profile of the 69-Bus system before and after reconfiguration (heavy load)

The optimum solution was reached after 170, 171 and 184 iterations respectively. The final HM of HMS=10 in case of normal load is given in (9). As shown, each row in the HM represents a radial configuration with the corresponding power losses

$$HM = \begin{bmatrix} 59 & 62 & 71 & 70 & 15: & 9.42 \\ 57 & 62 & 71 & 70 & 15: & 9.42 \\ 57 & 62 & 71 & 70 & 14: & 9.43 \\ 58 & 62 & 71 & 70 & 14: & 9.43 \\ 59 & 62 & 71 & 70 & 14: & 9.43 \\ 56 & 62 & 71 & 70 & 14: & 9.43 \\ 57 & 62 & 71 & 70 & 13: & 9.44 \\ 56 & 62 & 71 & 70 & 13: & 9.44 \\ 58 & 62 & 71 & 70 & 13: & 9.44 \\ 58 & 62 & 14 & 70 & 13: & 9.45 \end{bmatrix} \tag{9}$$

Table 2. Reconfiguration results of the 69-bus system (heavy load)

Reconfiguration	Losses (KW)	Tie Switches
Initial configuration	30.358	[10-70, 12-20, 14-90, 38-48, 26-54]
Final configuration using HS	Best 13.65 Worst 14.655 Average 13.67	[10-70, 12-20, 13-14, 47-48, 50-51]
Final configuration using SA [5]	13.65	[10-70, 12-20, 13-14, 44-45, 50-51]
Final configuration using SA [4]	13.65	[10-70, 12-20, 13-14, 45-46, 50-51]
Final configuration using MTS[6]	13.65	[10-70, 12-20, 13-14, 47-48, 50-51]

where the first row of the final harmony memory represents the optimum solution with the least power losses.

Fig. (3), (4) and (5) show the improvement of the voltage profile of the test system after reconfiguration in all loading conditions. Values of the lowest bus voltage (LBV) in pu before and after reconfiguration are listed in Table (4). The lowest bus voltage (LBV) was improved by 1.05%, 1.21%, and 0.484% for the normal, heavy and light loading respectively.

To verify the performance of the proposed HS algorithm, the system in the three loading cases was repeatedly solved for 50 runs. The best and the worst values among the best solutions as well as the average value of these 50 runs are also listed in the table corresponding to each loading condition. For the purpose of comparison, it is observed that, for the three loading patterns, the HS algorithm provides the same final power losses of that obtained by Simulated Annealing (SA), Tabu Search (TS) and Particle Swarm (PS) which indicates the effectiveness of the proposed algorithm.

Table 3. Reconfiguration results of 69-bus system (light load)

Reconfiguration	Losses (KW)	Tie Switches
Initial Configuration	5.0927	[10-70, 12-20, 14-90, 38-48, 26-54]
Final Configuration using HS	Best 2.32 Worst 2.4918 Average 2.3424	[10-70, 12-20, 13-14, 47-48, 50-51]
Final configuration using MTS[6] and SA [5]	2.32	[10-70, 12-20, 13-14, 44-45, 50-51]
Final configuration using SA [4]	2.32	[10-70, 12-20, 13-14, 45-46, 50-51]

Table 4. Values of LBV of initial and final configurations for the 3 loading conditions

	Normal load	Heavy load	Light load
LBV of initial configuration	0.972	0.9669	0.9865
LBV of optimum configuration	0.9824	0.9788	0.9913

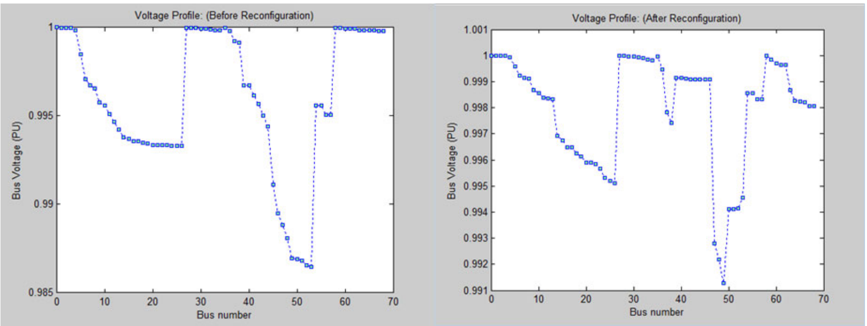


Fig. 5. Voltage profile of the 69-Bus system before and after reconfiguration (light load)

5 Conclusion

In this paper, a HS algorithm is proposed and successfully applied to solve the distribution network reconfiguration problem for loss reduction. The HS algorithm is a random search music-based technique. It does not require neither any initial value setting for the variables nor any prior domain knowledge such as gradient information of the objective function. It can consider discontinuous functions as well as continuous functions and can handle both discrete and continuous variables. HS is free from divergence and can escape local optima. The validity and effectiveness of the HS algorithm was demonstrated by applying it to a 69-bus system. The computational results obtained from solving the test system are investigated and compared to other previously applied algorithms available in literature. It is observed that, by nature, the harmony search incorporates the structure of existing heuristic methods. It is a population based approach similar to Ant Colony Optimization (ACO), it preserves the history of past vectors (harmony memory) similar to Tabu Search (TS), and is able to vary the adaptation rate (harmony memory considering rate) from the beginning to the end of computation resembling Simulated Annealing (SA), and manages several vectors simultaneously in a manner similar to Genetic Algorithm (GA). All these features make the HS algorithm more flexible than other heuristics.

References

1. Merlin, Back, H.: Search for a minimal-loss operating spanning tree configuration in an urban power distribution system. In: 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., Kim, J.-O., Shin, J.-R., Lee, K.Y.: An efficient simulated annealing algorithm for network reconfiguration in large-scale distribution systems. IEEE Trans. Power Del. 17(4), 1070–1078 (2002)
5. Chiang, H.D., Jean Jumeau, R.: Optimal Network Reconfigurations in Distribution Systems: Part 2. IEEE Transactions on Power Delivery 5(3) (1990)
6. 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)
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. Elec. Power Systems Research 31, 61–70 (1994)
8. Pan, Q.-K., Suganthan, P.N., Tasgetiren, M.F., Liang, J.J.: A Self-Adaptive Global Best Harmony Search Algorithm for Continuous Optimization Problems. Applied Mathematics and Computation 216(3), 830–848 (2010)
9. Pan, Q.-K., Suganthan, P.N., Liang, J.J., Tasgetiren, M.F.: A Local-Best Harmony Search Algorithm with Dynamic Subpopulations. Engineering Optimization 42(2), 101–117 (2010)
10. Shirmohammadi, D., Hong, H.W.: A compensation-based power flow method for weakly meshed distribution and transmission networks. IEEE Trans. Power Syst. 3 (1988)
11. Geem, Z.W., Kim, J.H., Loganathan, G.V.: A new heuristic optimization algorithm: Harmony search. Simulation 76, 60–68 (2001)
12. Geem, Z.W.: Music-Inspired Harmony Search Algorithm. Studies in Computational Intelligence 191 (2009)
13. Srinivasa Rao, R., Narasimham, S.V.L., Raju, M.R., Srinivasa Rao, A.: Optimal Network Reconfiguration of Large-Scale Distribution System Using Harmony Search Algorithm. In: IEEE Trans. on Power Systems (accepted for publication- article in press)

An Improved Multi-Objective Algorithm Based on Decomposition with Fuzzy Dominance for Deployment of Wireless Sensor Networks

Soumyadip Sengupta[1], Md. Nasir[1], Arnab Kumar Mondal[1], and Swagatam Das[2]

[1] Dept. of Electronics and Telecomm. Engg. Jadavpur University,
Kolkata - 700032, India
[2] Electronics and Communication Sciences Unit, Indian Statistical Institute,
Kolkata, India
{roni14may,nasir795,arnabkumarmondal123}@gmail.com,
swagatam.das@ieee.org

Abstract. The aim of this paper is to find a deployed sensor node arrangement to maximize the area of coverage, minimize the net energy consumption, maximize the network lifetime, and minimize the number of deployed sensor nodes maintaining connectivity between each sensor node and the sink node for proper data transmission. We have also assumed tree structure of communication between the deployed nodes and the sink node for data transmission. We have modeled the sensor node deployment problem as a multi-objective constrained problem maintaining all the above requirements. We have proposed a new fuzzy dominance based decomposition technique called MOEA/DFD and have compared its performance on other contemporary state-of-arts in multi-objective optimization field like MOEA/D and NSGAII. The algorithm introduces a fuzzy Pareto dominance concept to compare two solutions and uses the scalar decomposition method only when one of the solutions fails to dominate the other in terms of a fuzzy dominance level. MOEA/DFD performs better than all other algorithms.

Keywords: Multi-objective optimization, Fuzzy logic, Sensor network, Deployment.

1 Introduction

In this paper we have considered the deployment of sensor nodes[1,2] in a given area taking into account all the above mentioned objectives i.e. i) minimizing the number of sensor nodes to reduce cost and payload of deployment. ii) Minimizing the net energy consumed by all the nodes. iii) Maximizing the area covered by the nodes so that any event occurring in the region of interest is easily detected and the data can be sent to the sink node. iv) Maximizing the lifetime of the network. All the nodes deployed have certain limited initial energy. At each time cycle as the nodes transmit the received data to the sink node a certain amount of energy is consumed and the energy remaining in the sensor nodes decreases until it breaks down. Network lifetime can be thought of as

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 688–696, 2011.
© Springer-Verlag Berlin Heidelberg 2011

the time in which just one node breaks down or all the nodes fail thus hampering the whole network setup. v) Maintain connectivity in the deployed node configuration so every node can communicate with the sink node. The communication in the network is often structured as a tree graph, in which there is only a single path between each sensor node and the sink node. Maximizing coverage means the nodes must be placed far apart from the sink node, which is considered at the centre of the area concerned. Again minimizing energy consumption or maximizing network lifetime demands sensor nodes to be placed near the sink node. Thus it is evident that both the objectives are conflicting in nature and thus multi-objective technique is the best way to deal with those problems [3, 4, and 5]. Connectivity problem is modeled as a constraint problem using tree structure of communication. In this work, we have used a fuzzy dominance concept to synthesis a new algorithm based on the decomposition strategy of MOEA/D called MOEA/DFD (Decomposition with Fuzzy Dominance). Net energy consumed by the sensor nodes and the non-coverage of the demand points are modeled as the two objectives with connectivity as the constraint. Our goal is to minimize both the objectives to study the variation of lifetime and energy. From the obtained Pareto front we employ a decision making technique to select a particular node configuration with minimum energy above a defined non-coverage level. We have compared the result of our algorithm with that of MOEA/D and NSGAII. We have seen that MOEA/DFD surpasses NSGAII[6] and MOEA/D[7] in almost all the cases.

The rest of the paper is organized in the following way. In section 2, we model the sensor node deployment problem in a given area and formulate the necessary objective functions and constraints. In section 3, we discuss about the basics of the multi-objective problems (MOPs) and present the algorithm MOEA/DFD. In section 4, we describe the experimental set-up and parameters and present our results and figures with explanation. In section 5, we finally conclude the paper and unfold a few important future research issues.

2 Problem Definition

Let us consider a set of demand points. Each node have a definite radius of sensing, *R_sense* and any demand points within that circle of radius *R_sense* is said to be covered by that sensor node. The coverage problem in WSN requires that each demand point must be covered by at least one sensor node. On the other hand connectivity means at least one path must be there between each sensor node and the sink node. Two sensor nodes are said to be connected if the distance between them are less than *R_comm* (radius or range of communication the sensor nodes). The energy consumed in the sensor node can be primarily composed into three parts: i) maintenance energy ii) reception energy iii) transmission energy. Each node has a specific initial energy. Energy consumed by each node gives the measure of the working time of the node and thus the network lifetime. Here the decision variable is an array containing positions of the nodes in the plane.

The energy consumed by each node e_i can now be formulated as:

$$e_i = ME_i + TE_i \times P_{im} + RE_i \times \alpha_i, \qquad (1)$$

where, *ME* = node maintenance energy; *TE* = node transmission energy; *RE* = node reception energy; P_{im} means the cost of the minimum path from a sensor node i to the sink node m ; α_i refers to the number of sensor nodes from which the node i receives data and transfer it to the sink node in multi-hop communication.

Thus the net energy consumed and the objective function f_1 is defined as:

$$f_1 = \sum_{i \in S} e_i \,. \tag{2}$$

Similarly the two life-time T_1 and T_2 for the WSN breakdown are defined as :

$$T_1 = \min\{t_i\} \quad \text{and} \quad T_2 = \max\{t_i\} \qquad \forall i \in S \,. \tag{3}$$

The second objective is to maximize the area of coverage. But since we need to minimize energy, coverage must also be modeled as a minimization problem as:

$$f_2 = \frac{(\sum_{j \in D} h_j)}{|h_j|} \,, \tag{4}$$

Where h_j = variable to indicate if demand point j is not covered, assumes 1 when point j is not covered.

Thus the net multi-objective approach can be summed up as:

$$\min Y = F\left(\vec{x}\right) = (f_1, f_2) \tag{5}$$

$$NCP_i = 0$$

NCP is defined as a Non-Connectivity Penalty parameter.

We use of graph theory is to satisfy or fulfill three objectives:

i) Ensure connectivity in the WSN by assuming tree structure to check whether the sensor nodes are connected to the sink or not. ii) To find a minimum path for transmission of data from the sink node i to the sink node iii) To find the number of nodes α_i from which the node i receives data while considering the minimum path for each node as established before.

So Dijkstra's shortest path algorithm[8] is applied from each of the sensor node to the sink node.

3 Multi-Objective Optimization, MOEA/DFD – Brief Background

A multi-objective optimization problem (MOP) can be formally stated as follows:

$$\begin{array}{ll} \text{Minimize} & F(\vec{x}) = (f_1(\vec{x}),...,f_m(\vec{x})) \\ \text{subject to} & \vec{x} \in \Omega, \end{array} \quad (6)$$

where Ω is the decision space, $F : \Omega \rightarrow R^m$ consists of m real-valued objective functions and R^m is called the objective space. If $\vec{x} \in R^n$, all the objectives are continuous and Ω is described by $\Omega = \{\vec{x} \in R^n \mid h_j(\vec{x}) \leq 0, j = 1,...,k\}$, where h_j are continuous functions, we call (6) a continuous MOP.

The basic definition of dominance and Pareto optimality [9] played very important roles in the development of effective MOEAs. However, they do not provide a complete framework for easily implementing new methods. The reason is two-fold: Firstly, the basic definition of dominance does not make a difference between two solutions when neither is dominating, and secondly, it measures the extent by which one solution dominates the other. In order to circumnavigate these problems, some numerical measures for dominance have been proposed recently. In this article we shall use the concept of fuzzy dominance with respect to fuzzy-i-dominance by a solution and fuzzy dominance by a solution [10].

3.1 MOEA/DFD: A Brief Discussion

MOEA/DFD explicitly decomposes the MOP (6) into *N* scalar optimization subproblems during non domination with fuzzy dominance, where *N* is the population size of the algorithm. It solves these subproblems simultaneously by evolving a population of solutions. It is well-known that a Pareto optimal solution to a MOP could be an optimal solution of a scalar optimization problem in which the objective is an aggregation of all the f_i *'s,* where f_i *'s* denote the i^{th} objective function value, with variable uniformly spread weight vectors. There are several approaches for converting the problem of approximation of the Pareto Front into a number of scalar optimization problems. In this paper, we use the Tchebycheff approach method [2].

- A population of N points $\vec{x}^1, \vec{x}^2,..., \vec{x}^N \in \Omega$, where $\vec{x}^i$ is the current solution to the *i*-th subproblem;
- $FV^1,...,FV^N$, where FV^i is the F-value of $\vec{x}^i$, i.e. $FV^i = F(\vec{x}^i)$ for each $i = \{1,...,N\}$;
- $\vec{z} = (z_1,...,z_m)^T$, where z^i is the best(lowest) value found so far for objective f_i;
- *gen*: the current generation number; *max_gen* = maximum number of generation.
- A = fuzzy function decay constant; τ = domination threshold
- μ_i^j = fuzzy i-dominance level; $\hat{\mu}^j$ = fuzzy dominance level

The algorithm works as follows:

Input : i) MOP(1) ii) a stopping criteria iii) N the number of the subproblems considered in MOEA/DFD iv) a uniform spread of N weight vectors: $\lambda^1,\ldots,\lambda^N$ v) T : the number of the weight vectors in the neighborhood of each weight vector;

Output : $\{\vec{x}^{\,1},\ldots,\vec{x}^{\,N}\}$ and $\{F(\vec{x}^{\,1}),\ldots,F(\vec{x}^{\,N})\}$

Step 1 Initialization

Step 1.1: Compute the Euclidean distances between any two weight vectors and then find the T closest weight vectors to each weight vector to find $B(i)$

Step 1.2: Generate an initial population $\vec{x}^{\,1},\ldots,\vec{x}^{\,N}$ in the search space.

Step 1.3: Initialize $\vec{z}$ by setting $\vec{z}$ =min $\{f_i(\vec{x}^{\,1}),\ f_i(\vec{x}^{\,2}),\ldots,\ f_i(\vec{x}^{\,N})\}$.

Step 1.4: Set gen=0 for all $i=\{1,\ldots,N\}$.

Step 2 For each $i=\{1,\ldots,N\}$ do:

Step 2.1: Selection of mating/update range based on ∂ .

Step 2.2: Reproduction: Set $r_1=i$ and randomly select one indexes r_2 from P, and then generate a solution $\vec{y}$ from x^{r_1} and x^{r_2} by DE/rand/1/bin.

Step 2.3: Repair: If an element of $\vec{y}$ is out of the boundary reset its value.

Step 2.4: Update of z**:** For each $j=1,\ldots,m$, if $z_j > f_j(\vec{y})$, then set $z_j = f_j(\vec{y})$.

Step 2.5: Calculation of fuzzy dominance level:

For each $j\in P$,

For each $i\in\{1,\ldots,m\}$

if $f_i(\vec{y}) \le f_i(\vec{x}^{\,j})$, $\mu_i^j = 1$;

else $\mu_i^j = e^{-A(f_i(\vec{y})-f_i(\vec{x}^{\,j}))}$; end.

$\hat{\mu}^j = \cap\mu_i^j = \prod_{i=1}^{m}\mu_i^j$;end.

$\tau = 0.1 * gen/(\max\_gen)$;end.

Step 2.6: Update of solutions: For each $j\in P$; If $\hat{\mu}^j > \tau$, then $x^j = y$, $FV^j = F(\vec{y})$

else, if $g^{te}(\vec{y}^{\,j} \mid \lambda^j, \vec{z}^*) \le g^{te}(\vec{x}^{\,j} \mid \lambda^j, \vec{z}^*)$, then $\vec{x}^{\,j} = \vec{y}$, $FV^j = F(\vec{y})$;

Step 4 If the stopping criteria is satisfied then stop and output.

Step 5 $gen=gen+1$. Go to **Step 2.**

Though decomposition is a very effective process for maintaining convergence and diversity at the same time for a multiobjective optimization problem, still there are few drawbacks in this method which are mainly reflected in the convergence towards the Pareto front. Each of the weight vectors associated with an individual in the population tries to select a new offspring whose position in the objective function hyper space is farther away from the parent individual and hence in this process diversity is maintained. But in course of this process, the weight vector having affinity towards the offspring at a further position in the function space with respect to the parent individual, will neglect those offspring which may be potentially better but lying closer in comparison to that parent individual. Constraint handling technique for the above algorithm is that used by Deb *et.al.* [9].

4 Experimental Results

4.1 Experimental Setup

We consider a square area of 60m length in each side where the sensors are to be deployed for monitoring purpose. We haven consider 12×12=144 demand points in the whole area. We consider a setup of 10 nodes to be deployed in the area. Other parameters of the sensor networks are described below:

- R_sense = 15m; R_comm= 15m;
- *ME* = 13mA; *TE* = 20mA; *RE* = 2mA; Initial energy=1Ah
- *NCP* = 1 for each node being not connected with the sink node.

The sink node is placed at the middle of the area to be monitored.

Algorithm parameters: T=0.1N; ∂=0.9; A= 50; τ is varied from1 to 0.9 with increase in F.Es. Maximum function evaluation is 1, 00,000 with population of 500.

Other parameters are kept as recommended.

4.2 Results

MOEA/D, NSGAII and MOEA/DFD are simulated on the above described test bed and the Pareto front diagram is shown in Figure 1. To justify the performance of the multi-objective algorithms we use two performance indicator namely coverage metric [6] and spacing metric to judge the performance both in terms of convergence and diversity. Table 2. and Table 3. present the coverage metric and spacing metric results for the above mentioned algorithms on 10 node setup.

Table 1. Energy Consumptions and Lifetimes for 95% coverage

Spacing Node setup	MOEA/DFD	MOEA/D	NSGAII
Energy Consumption (in mA)	**3.7655**	3.8787	3.8354
Minimum Lifetime (in hour)	**1.6500**	1.6252	1.5880
Maximum Lifetime (in hour)	**19.6134**	7.6760	6.2664

Table 2. Coverage metric measure for MOEA/DFD (A), MOEA/D (B) and NSGAII(C)

C-metric Node setup	C(A,B)	C(B,A)	C(A,C)	C(C,A)
Node=10	0.5631	0.3167	0.8256	0.1372

Table 3. Spacing metric for MOEA/DFD, MOEA/D and NSGAII

Spacing Node setup	MOEA/DFD	MOEA/D	NSGAII
Node=10	**0.0032**	0.0089	0.0333

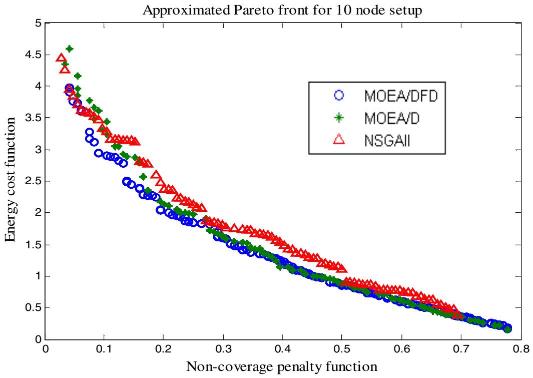


Fig. 1. Approximated Pareto front for 10 node setup

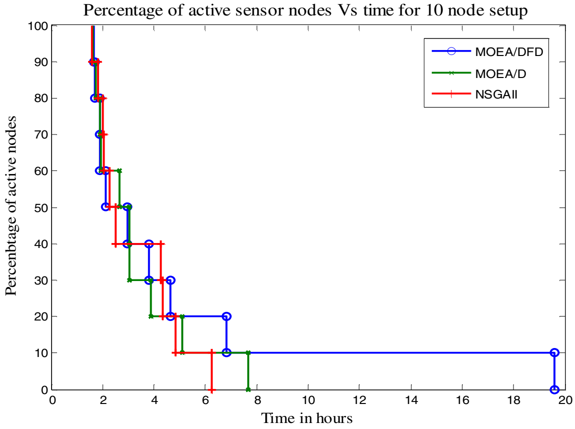


Fig. 2. Percentage of active nodes Vs time

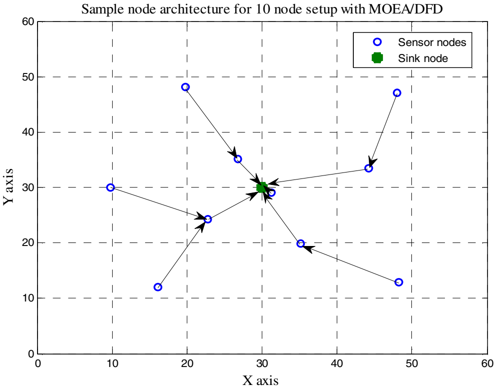


Fig. 3. Sample node architecture for 10 node setup

From the results presented we see that MOEA/DFD performs best in both cases and also Pareto front representation validates it. The main motivation for the designing of the deployment problem is the minimization of the energy consumption and maximization of the lifetime of the network presented in Table 3. The decision maker is employed to select the best compromise solution with minimum energy having 95% coverage from the Pareto front. Figure 2. presents the percentage of active nodes with time and Figure 3. represents a sample node architecture as obtained. MOEA/DFD is also best in all these instances.

5 Conclusion

This paper proposes a new fuzzy dominance concept which falls under a category of newly defined relaxation dominance concept. MOEA/DFD happens to tackle the multi-objective sensor node deployment problem better than other contemporary state-of-art. The deployment problem of sensor nodes can be modified and applied in areas where probabilistic event detection is necessary rather than distributed event detection. In recent times density control of a largely deployed sensor node through sleep scheduling is an important area of research. The multi-objective approach can also be extended to those areas for better flexibility and higher accuracy. So there are lots of opportunities in all those areas where evolutionary computation as an efficient optimizer had rarely been used.

References

1. Akyildiz, I., Su, W., Sankarasubramaniam, Y., Cayirci, E.: A survey on sensor networks. IEEE Communication Magazine, 102–114 (2002)
2. Liu, X., Mohaparta, P.: On the deployment of wireless data back-haul networks. IEEE Transaction on Wireless Communication 6(4), 1426–1435 (2007)

3. Zhou, A., Qu, B.Y., Li, H., Zhao, S.-Z., Suganthan, P.N., Zhang, Q.: Multiobjective Evolutionary Algorithms: A Survey of the State-of-the-art. Swarm and Evolutionary Computation 1(1), 32–49 (2011)
4. 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)
5. Zhao, S.Z., Suganthan, P.N.: Two-lbests Based Multi-objective Particle Swarm Optimizer. Engineering Optimization 43(1), 1–17 (2011)
6. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Trans. Evolutionary Computation 6(2), 182–197 (2002)
7. Zhang, Q., Li, H.: MOEA/D: A multi-objective evolutionary algorithm based on decomposition. IEEE Transactions on Evolutionary Computation 11(6), 712–731 (2007)
8. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numerische Mathematik 1, 269–271 (1959)
9. Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons (2001)
10. Koduru, P., Das, S., Welch, S., Roe, J.L.: Fuzzy Dominance Based Multi-Objective GA-Simplex Hybrid Algorithms Applied to Gene Network Models. In: Deb, K., et al. (eds.) GECCO 2004. LNCS, vol. 3102, pp. 356–367. Springer, Heidelberg (2004)

Application of Multi-Objective Teaching-Learning-Based Algorithm to an Economic Load Dispatch Problem with Incommensurable Objectives

K.R. Krishnanand[2], Bijaya Ketan Panigrahi[1], P.K. Rout[2], and Ankita Mohapatra[2]

[1] Department of Electrical Engineering, Indian Institute of Technology, Delhi, India
[2]Multi-Disciplinary Research Cell, Siksha 'O' Anusandhan University, Bhubaneswar, India
bkpanigrahi@ee.iitd.ac.in, pkrout_india@yahoo.com,
{krishkr09,ankita.cet1208}@gmailcom

Abstract. In this paper, a multiobjective teaching-learning-based optimization algorithm with non-domination based sorting is applied to solve the environmental/economic dispatch (EED) problem containing the incommensurable objectives of best economic dispatch and least emission dispatch. The address of the environmental concerns that arise in the present day due to the operation of fossil fuel fired electric generators and global warming requires the transformation of the classical single objective economic load dispatch problem into multiobjective environmental/economic dispatch problem. In the work presented a test system of forty units is taken with fuel cost and emission as two conflicting objectives to be optimized simultaneously. The mathematical model used considers practical upper and lower bounds applicable to the generators. The valve point loading of the generator is mimicked in the modeling to accommodate a more realistic system. The simulation result reveals that the proposed approach is a competitive one to the current existing methods for finding the best optimal pareto front of two conflicting objectives and has the better robustness.

Keywords: Environmental/Economic dispatch, Multi objective, Pareto optimal front, Non-dominated sorting, Teaching-learning-based optimization.

1 Introduction

The electrical power systems are expected to meet the continuous variation of power demand that happens over time. Essentially, to ensure economic operation of the power system, the scheduling of power generation is performed based on two significant tasks, unit commitment and economic dispatch, of which, later is the topic of research in this paper. The intent of the traditional economic dispatch is to allocate generation levels to various generators in the system so that the load demand is met in the most economic way. However, the optimum economic schedule obtained may not be suitable to follow, in case the environmental criteria are also considered. The utilities in the present scenario are expected to reduce their SO_2 and NO_X emissions [1].

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 697–705, 2011.
© Springer-Verlag Berlin Heidelberg 2011

Therefore, apart from the objective of cost, lesser emission objective must also be taken into account.

Environmental/Economic dispatch (EED) is a multiobjective problem having conflicting objectives, since the minimization of cost tends to maximize the pollution in general. This leads to the requirement of trade-off analysis so as to define suitable dispatch policies for various levels of the demand [2]. There has been much research done regarding the EED problem. The influence of power pools on power dispatch with environmental consideration have been studied by taking emissions as the constraints of the model and power is dispatched by minimizing cost as single objective [3]. The multi-objective problem considering cost, emission and line over load index as objectives is solved using ε-constrained method in [4]. Over the past decade, the simultaneous handling of many objectives has received much interest due to the development of a number of multiobjective evolutionary search strategies [5]. The novel Non-dominated Sorting Genetic Algorithm (NSGA) [6], Niched Pareto Genetic Algorithm (NPGA) [7] and Strength Pareto Evolutionary Algorithm (SPEA) [8] was successfully applied to EED problem. An elitist multi-objective evolutionary algorithm called NSGA-II was developed in [9], [10] and applied to EED problem. In this paper a new multiobjective algorithm with an effective strategy is used to obtain the pareto curve of optimal solutions for a 40 units system given in [11].

Teaching-learning-based optimization (TLBO) is a very new algorithm introduced in [12]. This single objective algorithm is based on the effect of influence of a teacher on the output of the learners in the class. The teacher is considered as a highly learned person who does knowledge sharing with the learners in the class. The quality of the teachers exhibits its impact on the outcome of the learners, which is seen from their results or grades. In this work the main objective is to modify the TLBO algorithm to find pareto optimal solutions in a multiobjective problem domain. In this paper a multiobjective teaching-learning-based optimization (MOTLBO) algorithm using non dominated sorting procedure is developed, which is then applied to EED problem of 40 units system. The test system has the effect of valve point loading in its fuel cost characteristics [11].

The following part of the paper is organized as follows. In section 2 the objective is outlined along with the formulations followed in the system modeling. Section 3 describes the formulation of non-dominated sorting based MOTLBO algorithm. Section 4 describes the simulation strategy for implementing the solution to EED problem and the experimental results obtained. Section 5 is the conclusion.

2 EED Problem Formulation

The environmental economic dispatch being multi-objective has more than one objective against the optimization of each other. In this work, two objectives are considered. Both equality and inequality constraints are imposed while optimizing the two objectives of fuel cost and emission simultaneously.

$$Minimize\left[F\left(P_G\right), E\left(P_G\right)\right] \tag{1}$$

2.1 Fuel Cost Objective

The primary objective of the conventional economic dispatch is the minimization of total generation cost while satisfying several constraints. Mathematically expressed as

$$Minimize \quad F(P_G) = \sum_{i=1}^{N_G} F_i(P_{Gi}) \tag{2}$$

where $F_i(P_{Gi})$ is the i^{th} generator cost function and is approximated as a quadratic function of the power output from the generating units, i.e.

$$F_i(P_{Gi}) = a_i + b_i P_{Gi} + c_i P_{Gi}^2 \qquad \mathrm{i} = 1,2,3,\ldots,\mathrm{N_G} \tag{3}$$

where a_i, b_i, c_i are the cost coefficients, N_G represents the number of generating units in the system and P_{Gi} is the power output of the i^{th} generator. The generator cost function with non-smooth cost characteristics due to the valve point effect is given by

$$F_i(P_{Gi}) = a_i + b_i P_{Gi} + c_i P_{Gi}^2 + \left| e_i \sin\left(f_i \left(P_{Gi}^{\min} - P_{Gi} \right) \right) \right|, \qquad \mathrm{i} = 1,2,3,\ldots,\mathrm{N_G} \tag{4}$$

where e_i, f_i are the cost coefficients related to the valve-point loading and they contribute to the non-smoothness of the cost curve.

2.2 Emission Objective

The minimum emission dispatch is the simultaneous minimization of classical economic dispatch including emission objective which is modeled as:

$$E(P_G) = \sum_{i=1}^{N_G} \left(\alpha_i + \beta_i P_{Gi} + \gamma_i (P_{Gi})^2 + \zeta_i \exp(\lambda_i P_{Gi}) \right) \tag{5}$$

Here, $\alpha_i, \beta_i, \gamma_i, \zeta_i \; and \; \lambda_i$ are the emission coefficients of the i^{th} generator.

2.3 Power Balance Constraint

The total power generated must cover total demand P_D and total transmission losses P_{LOSS}. Therefore,

$$\sum_{i=1}^{N_G} P_{Gi} - P_D - P_{Loss} = 0 \tag{6}$$

Transmission losses are considered to be negligible in this system and no loss coefficients are given in [11].

2.4 Generation Capacity Constraints

The real output power of each generator is constrained by lower and upper limits.

$$P_{Gi}^{min} \le P_{Gi} \le P_{Gi}^{max} \tag{7}$$

where, P_{Gi}^{min} and P_{Gi}^{max} are the minimum and maximum operating outputs of unit *i* respectively.

3 Multiobjective Teaching-Learning-Based Optimization (MOTLBO) Algorithm

The TLBO algorithm is a very new algorithm recently introduced in [12]. This optimization technique performs based on the dependency of the learners in a class on the quality of teacher in the class. The teacher raises the average performance of the class and shares the knowledge with the rest of the class. The individuals are free to perform on their own and excel after the knowledge is shared. The whole procedure of TLBO is divided in to two phases, the Teacher phase and the Learner phase.

Initialization

Initially, a matrix of *N* rows and *D* columns is initialized with randomly generated values within the search space. The value *N* represents the population size of the "class size" in this case. The value *D* represents the number of "subjects or courses offered", which is same as the dimensionality of the problem taken. The procedure being iterative is set to run for *G* number of generations. The j^{th} parameter of the i^{th} vector (learner) in the initial generation is assigned values randomly using the equation

$$x_{(i,j)}^{1} = x_{j}^{min} + rand_{(i,j)} \times \left(x_{j}^{max} - x_{j}^{min}\right) \tag{8}$$

Where $rand_{(i,j)}$ represents a uniformly distributed random variable within the range (0,1). The parameters of the i^{th} vector (or learner) for the generation *g* are given by

$$X_{(i)}^{g} = \left\lfloor x_{(i,1)}^{g}, x_{(i,2)}^{g}, \ldots, x_{(i,j)}^{g}, \ldots, x_{(i,D)}^{g} \right\rfloor \tag{9}$$

The objective values at a given generation form a column vector. In a dual objective scenario, such as this one, two objective values are present for the same row vector. Two objectives (*a* and *b*) can be evaluated as

$$\begin{bmatrix} Ya_i^g \\ Yb_i^g \end{bmatrix} = \begin{bmatrix} fa\left(X_{(i)}^{g}\right) \\ fb\left(X_{(i)}^{g}\right) \end{bmatrix} \tag{10}$$

For all the equations used in the algorithm, $i=1,2,\ldots,N$, $j=1,2,\ldots,D$ and $g=1,2,\ldots,G$. The random distribution followed by all the *rand* values is the uniform distribution.

Teacher Phase

The mean vector containing the mean of the learners in the class for each subject is computed. The mean vector M is given as

$$M^g = \begin{bmatrix} mean\left(\left[x^g_{(1,1)},\ldots,x^g_{(i,1)},\ldots,x^g_{(N,1)}\right]\right) \\ \ldots \\ mean\left(\left[x^g_{(1,j)},\ldots,x^g_{(i,j)},\ldots,x^g_{(N,j)}\right]\right) \\ \ldots \\ mean\left(\left[x^g_{(1,D)},\ldots,x^g_{(i,D)},\ldots,x^g_{(N,D)}\right]\right) \end{bmatrix}^T \tag{11}$$

which effectively gives us

$$M^g = \left[m^g_1, m^g_2, \ldots, m^g_j, \ldots, m^g_D\right] \tag{12}$$

The best vector with the minimum objective function value is taken as the teacher ($X^g_{Teacher}$) for that iteration. The algorithm proceeds by shifting the mean of the learners towards its teacher. A randomly weighted differential vector is formed from the current mean and the desired mean vectors and added to the existing population of learners to get a new set of improved learners.

$$Xnew^g_{(i)} = X^g_{(i)} + rand^g \times \left(X^g_{Teacher} - T_F M^g\right) \tag{13}$$

where T_F is a teaching factor which is randomly taken at each iteration to be either 1 or 2. The superior learners in the matrix *Xnew* replace the inferior learners in the matrix *X* using the non-dominated sorting algorithm [10].

Learner Phase

This phase consists of the interaction of learners with one another. The process of mutual interaction tends to increase the knowledge of the learner. Each learner interacts randomly with other learners and hence facilitates knowledge sharing. For a given learner, $X^g_{(i)}$ another learner $X^g_{(r)}$ is randomly selected ($i \neq r$). The i^{th} vector of the matrix *Xnew* in the learner phase is given as

$$Xnew_{(i)}^{g}=\begin{cases} X_{(i)}^{g}+rand_{(i)}^{g}\times\left(X_{(i)}^{g}-X_{(r)}^{g}\right) & if\left(Y_{i}^{g}<Y_{r}^{g}\right)\\ X_{(i)}^{g}+rand_{(i)}^{g}\times\left(X_{(r)}^{g}-X_{(i)}^{g}\right) & otherwise \end{cases} \tag{14}$$

The MOTLBO algorithm, due to the multiobjective requirements, adapts to the scenario by having multiple *Xnew* matrices in the learner phase, one for each objective. So, the learner phase operations for a dual objective problem are as shown in equations below.

$$\left.Xnew_{(i)}^{g}\right|_{a}=\begin{cases} X_{(i)}^{g}+rand_{(i)}^{g}\times\left(X_{(i)}^{g}-X_{(r)}^{g}\right) & if\left(Ya_{i}^{g}<Ya_{r}^{g}\right)\\ X_{(i)}^{g}+rand_{(i)}^{g}\times\left(X_{(r)}^{g}-X_{(i)}^{g}\right) & otherwise \end{cases} \tag{15}$$

$$\left.Xnew_{(i)}^{g}\right|_{b}=\begin{cases} X_{(i)}^{g}+rand_{(i)}^{g}\times\left(X_{(i)}^{g}-X_{(r)}^{g}\right) & if\left(Yb_{i}^{g}<Yb_{r}^{g}\right)\\ X_{(i)}^{g}+rand_{(i)}^{g}\times\left(X_{(r)}^{g}-X_{(i)}^{g}\right) & otherwise \end{cases} \tag{16}$$

The *X* matrix and the *Xnew* matrices are passed together to the non-dominated sorting algorithm and only *N* best learners are selected for the next iteration.

Algorithm Termination

The algorithm is terminated after *G* iterations are completed. The final set of learners represents the pareto curve through their objective values.

4 Simulation Results

The simulations have been done using MATLAB software package on a Core2 Duo Intel processor with 3 GHz clock speed and supported by 3 GB of random access volatile memory. The proposed procedure was executed a few times out of which the best solution set is presented here. As an advantage, this algorithm has no parameters to be tuned and hence exhibits homogeneous behavior in all cases, without any requirement of parametric study of the algorithm.

The simulation strategy constitutes of estimating the solutions contributing to the extreme points in the expected pareto curve and then performing the multiobjective evaluations with those solutions points as the initial search points in the MOTLBO algorithm. The optimization capability of the algorithm makes the estimated pareto move nearer to the desired pareto as the iteration progresses. In this current study, only two points, one from each extreme of the pareto curve, are estimated by the TLBO algorithm and used as the initial seeds for the MOTLBO algorithm. The rest of the learners are initialized randomly.

Table 1. Generation (MW), cost ($\times 10^5$ \$) and emission ($\times 10^5$ ton) of 40 units system for a power demand of 10,500 MW

Unit	TLBO		DE [11]	
	Economic Dispatch	Emission Dispatch	Economic Dispatch	Emission Dispatch
1	112.2660	114.0000	110.9515	114.0000
2	114.0000	114.0000	113.2997	114.0000
3	120.0000	120.0000	98.6155	120.0000
4	179.7330	169.3680	184.1487	169.2933
5	90.4253	97.0000	86.4013	97.0000
6	140.0000	124.2574	140.0000	124.2828
7	300.0000	299.7114	300.0000	299.4564
8	284.6000	297.9149	285.4556	297.8554
9	284.6000	297.2601	297.5110	297.1332
10	130.0000	130.0000	130.0000	130.0000
11	94.0000	298.4101	168.7482	298.5980
12	168.8000	298.0260	95.6950	297.7226
13	125.0000	433.5576	125.0000	433.7471
14	394.2790	421.7284	394.3545	421.9529
15	394.2790	422.7797	305.5234	422.6280
16	394.2790	422.7797	394.7147	422.9508
17	489.2790	439.4129	489.7972	439.2581
18	489.2790	439.4029	489.3620	439.4411
19	511.2790	439.4129	520.9024	439.4908
20	511.2790	439.4129	510.6407	439.6189
21	523.2790	439.4464	524.5336	439.2250
22	523.2790	439.4464	526.6981	439.6821
23	523.2790	439.7721	530.7467	439.8757
24	523.2790	439.7721	526.3270	439.8937
25	523.2790	440.1118	525.6537	440.4401
26	523.2790	440.1118	522.9497	439.8408
27	10.0000	28.9937	10.0000	28.7758
28	10.0000	28.9937	11.5522	29.0747
29	10.0000	28.9937	10.0000	28.9047
30	97.0000	97.0000	89.9076	97.0000
31	190.0000	172.3319	190.0000	172.4036
32	190.0000	172.3319	190.0000	172.3956
33	190.0000	172.3319	190.0000	172.3144
34	164.8010	200.0000	198.8403	200.0000
35	164.8000	200.0000	174.1783	200.0000
36	165.0640	200.0000	197.1598	200.0000
37	110.0000	100.8384	110.0000	100.8765
38	110.0000	100.8384	109.3565	100.9000
39	110.0000	100.8384	110.0000	100.7784
40	511.2837	439.4125	510.9751	439.1895
Fuel Cost	121,468.47	129,954.64	121,836.98	129,956.10
Emission	364,048.19	176,682.27	374,790.57	176,683.27

Table 2. Cost (×10 $) and emission (×10 ton) comparison of different methods

Method	Fuel Cost	Emission
SPEA 2 [11]	125,807.69	211,097.77
NSGA-II [11]	125,825.20	210,949.08
PDE [11]	125,730.95	211,765.46
MODE [11]	125,792.20	211,190.23
MOTLBO (proposed)	**125,507.37**	211,644.62

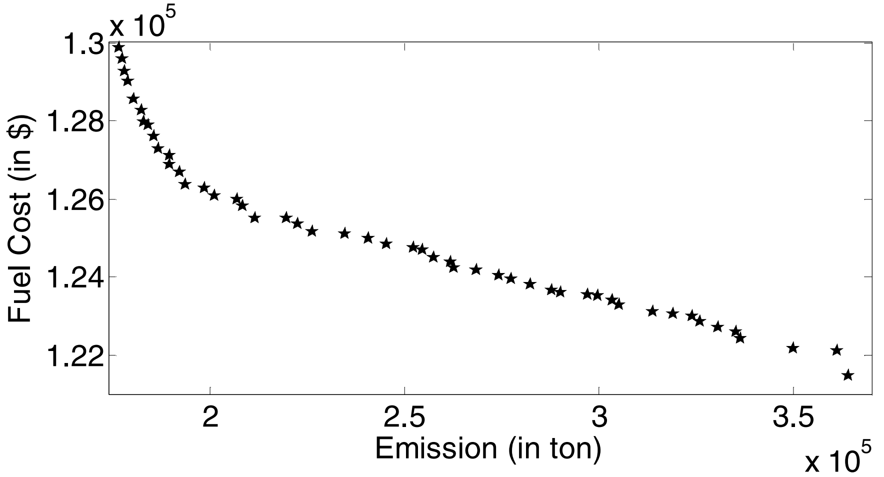


Fig. 1. Pareto-optimal curve for the last generation of MOTLBO

The results given in Table 1 shows that the MOTLBO algorithm with the strategy mentioned performs far better than the algorithms reported in recent literature [11]. The Fig 1 shows the capability of the proposed algorithm in achieving the pareto curve. Table 2 shows the best compromised solutions of different algorithms. The results indicate that MOTLBO algorithm can be used as a suitable technique for EED class of problems.

5 Conclusion

The EED problem consisting of 40 thermal generating units has been taken and modeled as per the literature available. TLBO algorithm has been adapted to solve multiobjective problems. The MOTLBO algorithm has been successfully applied to the test system considered. The procedure followed has resulted in superior results as compared to recent research literature. The feasibility and the suitability of the proposed algorithm have been verified. The procedure discussed is easily extensible to similar problems without any requirement of tuning of parameters.

References

[1] IEEE Current Operating Problems Working Group. Potential impacts of clean air regulations on system operations 10(2), 647–656 (1995)

[2] Zahavi, J., Eisenberg, L.: An application of the Economic-environmental power dispatch. IEEE Trans. Syst., Man, Cybernet. SMC-7(7), 523–530 (1977)

[3] Brodesky, S.F., Hahn, R.W.: Assessing the influence of power pools on emission constrained economic dispatch. IEEE Trans. Power Systems 1(1), 57–62 (1986)

[4] Yokoyama, R., Bae, S.H., Morita, T., Sasaki, H.: Multiobjective optimal generation dispatch based on probability security criteria. IEEE Trans. Power Systems 3(1), 317–324 (1988)

[5] Deb, K.: Multiobjective Optimization Using Evolutionary Algorithms. John Wiley & Sons, Chichester (2001)

[6] Abido, M.A.: A Novel Multiobjective Evolutionary Algorithm for Environmental / Economic Power Dispatch. Electric Power Systems Research 65(1), 71–81 (2003)

[7] Abido, M.A.: A Niched Pareto Genetic Algorithm for Environmental/ Economic Power Dispatch. Electric Power Systems Research 25(2), 97–105 (2003)

[8] Abido, M.A.: Environmental / Economic Power Dispatch using Multiobjective Evolutionary Algorithms. IEEE Trans. Power Systems 18(4), 1529–1537 (2003)

[9] Ah King, R.T.F, Rughooputh, H.C.S., Deb, K.: Evolutionary Multi-Objective Environmental/Economic Dispatch: Stochastic vs. Deterministic Approaches. KanGAL Report Number-2004019, 1–15 (2004)

[10] Ah King, R.T.F, Rughooputh, H.C.S., Deb, K.: Stochastic Evolutionary Multiobjective Environmental/Economic Dispatch. In: Proceedings of IEEE Congress on Evolutionary Computation, Vancouver, BC, Canada, pp. 946–953 (2006)

[11] Basu, M.: Economic environmental dispatch using multi-objective differential evolution. Applied Soft Computing 11(2), 2845–2853 (2011)

[12] 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(3), 303–315 (2011)

Application of NSGA – II to Power System Topology Based Multiple Contingency Scrutiny for Risk Analysis

Nalluri Madhusudana Rao[1], Diptendu Sinha Roy[2], and Dusmanta K. Mohanta[3]

[1] School of Computing, Sastra University, Thanjavur, India
[2] Department of Computer Science and Engineering,
National Institute of Science and Technology, Berhampur, India
[3] Department of Electrical and Electronics Engineering,
Birla Institute of Technology, Mesra, India
{madhu031083,diptendu.sr}@gmail.com, dkmohanta@bitmesra.ac.in

Abstract. The Incorporation of deregulation and increase in renewable sources of generation has shifted the nature of existing power systems to a more geographically distributed system. This had led to significant challenges towards on-line monitoring and control. Contingency set identification is an essential step in monitoring the power system security level. Multiple contingency analysis forms the basis of security issues, particularly of large, interconnected power systems. The difficulty of multiple contingency selections for on-line security analysis lies in its inherent combinatorial nature. In this paper, an approach for identification of power system vulnerability to avoid catastrophic failures is put forward, as a multi objective optimization problem that partitions its topology graph, accounts for maximizing the imbalance between generation and load in each island and at the same time minimizes the number of lines cut to realize the partitions. The Nondominated Sorted Genetic Algorithm, version II (NSGA II) has been applied to obtain the optimal solutions and the methodology involved has been applied to an IEEE 30 bus test system and results are presented.

1 Introduction

With the increased dependence of electric power in all aspects of modern society, power system reliability has become an area of prime importance. Electric power system is prone to much vulnerability, some of which are frequently encountered, whereas others are relatively erratic. Identifying small groups of lines, whose removal would lead to cascading failures, even a severe blackout, is critical for the secure operation of the electric power grid. The causes of cascading events in power systems are various [1]. One major contribution to cascading is high order initiating contingencies—removal of several power system components in a very short time, typically within seconds. Contingency set identification is an essential step in monitoring the power system security level [1]. Most literature [2] on contingency selection emphasizes screening methods to select contingencies from a presumed N−1 contingency set plus a limited number of high order contingencies, ranking them using an appropriate severity index.

There are a few related literatures that deal with systematic analysis of higher order, multiple contingencies, like [3], [4]. Additionally, such analysis requires huge amount of computations that make them ineffective for online deployment purposes.

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 706–713, 2011.
© Springer-Verlag Berlin Heidelberg 2011

Such non-ordered contingencies may lead to catastrophes and control actions must be taken to avoid such failure [5]. One approach to deal with this situation is to divide the electric power network into smaller sub-networks, called islands. The basis for forming the islands is to minimize the imbalance between generation and load in each island [4], [6].

Available literature on systematic selection of high order contingencies, called N−k contingencies (where $k \geq 2$) is limited. [5] and [6] proposed the on-line detection of hidden failure in protection device to prevent cascading failure. The proposed method needs exhaustive information on the logic of protection device installed in power system, which make it very difficult to be implemented. One might argue that most of these contingencies are so low in probability that they do not warrant attention. However, N−k contingencies do occur, and when they do, effects can be very severe, and these very pragmatic facts motivate the research in the field to identify high risk N-k contingencies for on-line security analysis [1].

This paper deals with the vulnerability assessment of the electric power grid. It formulates the identification of a small group of lines (multiple contingency), as a multi objective optimization problem taking into account the power imbalance among the different islands formed as well as the number if lines to be cut, such that when removed, they lead to a catastrophe. A multi objective optimization method, elitist Nondominated Sorting Genetic Algorithm, version II (NSGA II) has been employed to the vulnerability analysis. The methodology has been applied to the test case of an IEEE 30 bus test system and results are presented.

The remainder of this paper is organized as follows: Section II gives a brief review of power system vulnerability analysis using graph theoretic approach. Formulation of power grid vulnerability analysis as a multi objective optimization function has been presented in Section III along with its NSGA II implementation. Section IV provides the detailed methodology with its application to the IEEE 30 bus test system. Section VI concludes the paper.

2 Power System Vulnerability Analysis: A Graph Theoritic Approach

Islanding refers to the condition of a distributed generation continuing to power location even though power from the electric utility is no longer present [4]. In this paper islanding refers to one or several distributed power system units forming an island. Load can consist of either an industrial plant or consumer facilities in a distribution grid. Several approaches can be used to separate the whole network into small systems. Classical powers systems usually consist of buses and transmission lines which connect each other. The power flows in transmission lines have different values and directions. Therefore, power system networks can be considered as a directed graph with different weights at vertices [6].

To delve into the graph theoretic approach, a small four node, five branch reference graph paraphrased from [6] has been taken as shown in figure 1.

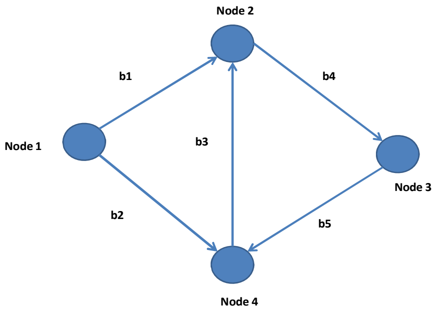


Fig. 1. A 4 reference graph with nodes and 5 branches

The four nodes and five branches are represented by circles and arrows respectively. Considering the direction of arrows a branch/node incidence matrix ‘A’ has been formed. So by using information A is given by

$$A = \begin{pmatrix} 1 & -1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{pmatrix} \tag{1}$$

where the rows represent the branches and the columns represent the nodes. Matrix entry A_{mn} is equal to1 if branch m originates at node n, and it is equal to -1 if the branch terminates to node n. The entry is equal to zero if the branch is not incident or connected to n. So from this point of view, two groups of values have been taken which is called indicator vector (X) to specify the partition.

$$X = [-1 \quad -1 \quad 1 \quad 1\,]^{T} \tag{2}$$

Multiplying the incidence matrix by the indicator vector yields a vector(Y) that identifies the branches that separate the two groups:

$$Y = AX = \quad [0 \quad -2 \quad 2 \quad 2 \quad 0]^{T} \tag{3}$$

The value of an entry in vector *y* is 0 if the branch is internal to a group and is ±2 if the branch connects the two groups. The sign depends on the direction of the arrow. From (3) it is obvious that

$$Y^{T} Y = X^{T}(A^{T} A)\, X = 4C \tag{4}$$

Where, C= number of separating branches. Here A^{T} A is called Laplacian matrix associated with the graph. It is denoted by L. From that Laplacian matrix, eigen value can be generated. The smallest positive eigen value of the Laplacian matrix and corresponding eigenvector are often referred to as Fiedler eigen value and Fiedler eigenvector [6], which plays a vital role in partitioning graphs by spectral approach [6]. From that reference graph, Laplacian matrix cab be derived. Now, power system will lead to an extreme event when some specific line outages lead to severe imbalance in the system. The focus of this paper is to identify a small number of lines that if removed will partition the system and will cause a severe power imbalance. These are two possibly competing objectives: Minimize cuts; and Maximize directed power flow across the cut set.

To proceed, first of all a description of these objectives is needed in terms of an indicator vector that will describe the partitions. The cut size cost is expressed by (4). Let the vector p denote the bus power injections and let x be an indicator vector with entries equal to ±1. Then, assuming a lossless network leads to:

$$p^{T}x = 2 \text{ (power flow between groups)} \tag{6}$$

The simplest, and arguably the most direct, mathematical representation for the objectives stated above is given by the following minimization problem:

$$x^{*} = \underset{x \in \{-1,1\}}{\arg} \ \text{Minimize } X^{T} L X - c\, p^{T} X \tag{7}$$

where the first term is equal to 4 times the number of cuts, the second term is proportional to the power flow disrupted by such cuts, and the positive scalar c explicitly represents the trade-off between these objectives. A small value for c favors minimizing cuts, while a large value of c favors maximizing power imbalance.

$$\text{Let } p = [2 \quad -2 \quad 1 \quad -1]^{T} \tag{8}$$

The n – 1 = 3 candidate indicator vectors are intuitively:

$$X1 = [1 \ \ 1 \ \ 1 \ \ -1]^{T} ; X2 = [1 \ \ -1 \ \ 1 \ -1]^{T} ; X3 = [1 \ \ -1 \ \ -1 \ \ -1]^{T} \tag{9}$$

For values of c less than 2, the minimum objective is obtained with $x3$. For values of c greater than 2, the minimum objective is obtained with $x2$. These are clearly the correct solutions of (7) for this example. For the small c case, more emphasis is placed on small cuts. The $x3$ solution contains the fewest cuts possible, 2, and has a greater power imbalance than $x1$, which also has 2 cuts. For large c, the power imbalance is more emphasized, and the indicator vector given by $x2$ maximizes this quantity in this example [6].

3 Multi Objective Formulation of Power System Risk Analysis: NSGA II Implementation

To make a power system N - k secure means that no combination of k facility outages will cause a wide spread blackout. The brute force method to test k components involves choosing all combinations of k outages and testing whether a blackout might occur .This is computationally impractical. The computation grows combinatorial as the size of the system increases .This is also one of the NP-Complete problems.

Multi-objective formulations are realistic models for many complex engineering optimization problems. The solution of such problems are not unique, but a set of equally efficient ,non-inferior or non dominated solutions known as Pareto- optimal set [8]. Evolutionary algorithms are popular approaches to solving multiobjective optimization. Currently most evolutionary optimizers apply Pareto-based ranking schemes [8]. Genetic algorithms such as the Non-dominated Sorting Genetic Algorithm-II (NSGA-II) have become standard approaches. The main advantage of evolutionary algorithms, when applied to solve multi-objective optimization problems, is the fact that they typically optimize sets of solutions, allowing computation of an approximation of the entire Pareto front in a single algorithm run.

Non-dominated sorted genetic algorithm, version II (NSGA II) has been particularly popular among its counterparts [8], [9].

This multiple contingency problem has been framed as multi-objective optimization problem and NSGA II has been employed for finding out the optimal solution. There are the two objective functions for this problem as shown in (11). One for minimizing the cut set and another for maximizing the power imbalance between the two groups. The first term of (11), namely ($0.5 * X^T L X$) denotes the number of lines cut; whereas the last term signifies the power imbalance between the two groups as has been mentioned in (7). It formally looks like this :

$$\text{Minimize} \qquad 0.5 * X^T L X - c * 0.25 * P^T X \tag{11}$$

and are constrained by two constraints namely$0.5 * X^T L X \leq C$; and $0.25 * P^T X \geq P$;

where C is the cut set count and P is the power imbalance between the two groups and X is a vector consisting of either -1 and/or +1 and of size equal to the number of nodes in the graph. The NSGA II algorithm used is as follows:

Step1: Initialize parent population of size N individuals

Step 2: Evaluate objectives and constraints for all individuals in parent population, rank the population and compute crowding distance.

Step 3: Use non-domination criterion based selection operator to identify high fitness individuals.

Step 4: Apply crossover and mutation operators to create child population.

Step 5: Evaluate all individuals in child population.

Step 6: Combine parent and child populations ,rank the combined population and compute the crowding distance.

Step 7: Select the best individuals to fill the new parent population of size N.

Step 8: Repeat steps 3 to 7 till the convergence or maximum number of generations.

Computational complexity for finding the no domination criteria is $O(MN^2)$,where M is the number of objective functions and N is the population size [9]. The following section uses this problem formulation and applies it to a standard power system test case, namely the IEEE 30 bus test system [10] and presents the results.

4 Case Study and Results

This section presents the detailed methodology involved to identify power system vulnerability taking the topology graph of power system. To find an optimal combination of power imbalance and line cuts, which are two contending features, NSGA II has been applied. The efficacy of the optimization employed has been demonstrated by means of a case study that considers the IEEE 30 bus test case, a system widely studied in literature. The IEEE 30 Bus Test Case [10] represents a portion of the American Electric Power System. The data has been entered in IEEE Common Data Format since 1993. In this system 5 generators are there. This paper islands the same using graph theory approach considering the power injections at different buses. Corresponding graph of IEEE 30 bus system graph has been shown in figure 2. The bus power injection values are also taken from the original IEEE 30 bus system data [10].

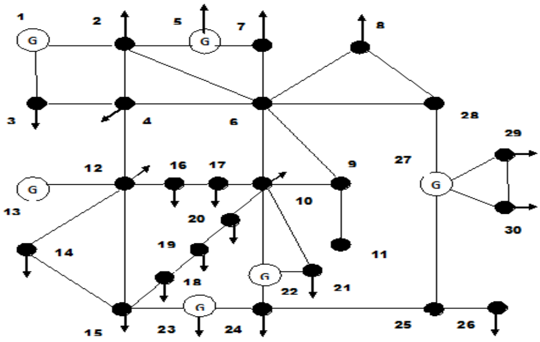


Fig. 2. System Graph of IEEE 30 bus System

For the 30 bus system, the A vector in (1) will have 30 X 41 entries, which could not be shown due to space limitations. Thereafter it has been partitioned into two islands from which appropriate indicator vector has to be supplied. It has to be noted that the indicator vector for 30 bus system the same structure as (2), with a total of 30 entries. For two islands, it will have two distinct entries respectively, i.e., -1 and 1 as per (2).

The NSGA II algorithm has been run on a desktop computer having the following details: Processor- Intel (R) Core(TM) 2 Duo CPU E7400@2.80GHz, 2048MB main memory, CPU Core Count- 2, Memory Bus Speed- 800MHz, Hard Drive- 320GB, D-Link Wireless GDWA- 510 Desktop Adapters having Ubuntu Linux 10.0 as host operating system.

Power and bus injection table has been taken from [6]. According to the data of that table, the p vector can be formed, that has 30 entries. While applying NSGA II, a case is presented considering different boundary conditions as follows:

$$0.5 * X^T \; L \; X \leq 11$$

$$0.25 * P^T \; X \geq 1150$$

where X is a vector of -1 and/or +1, of size 30. After solving it using NSGA-II, the results obtained has been tabulated in Table 1. The line cuts dissects the system graph into two disjoint partitions as is shown in figure 3.

Table 1. The two Optimal partitions obtained with Maximum Link cuts limited to 11 and minimum power imbalance chosen as 1150

Island 1	Island 2	Cutset	Power Imbalance
{1,13,22 ,23,25,27,2 9,30}	{2,3,4,5,6,7,8,9,10, 11,12,13,14,15,16,17,1 8,19,20,21,24,26,28}	{(27,28),(23,15),(13,12),(22, 10),(21,22),(24,25),(24,23),(22,2 4),(26,25),(1,2),(1,3)}	1276.2 unit

A note on the Pareto optimal front of the problem formulation using the proposed NSGA II scheme is provided as follows: The complete order over individuals is found out [11],[12]. Front 1: The individuals which are not dominated by any one of the individual in both the objective functions. Front 2 : The individuals which are not dominated by any one of the individual in only one objective function. Front 3: The individuals which are dominated by Front1. Front 4: The individuals which are dominated by Front2**.** Front 5: Rest of the individuals which are not coming under front 1, Front2, Front3 and Front4**.**

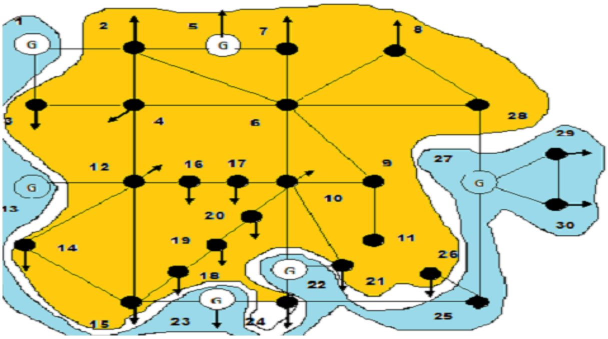


Fig. 3. Optimal partitions obtained of the graph of fig. 2 with Maximum Link cuts limited to 11 and minimum power imbalance chosen as 1150

5 Conclusion

This paper presents an approach for identification of power system vulnerability to avoid catastrophic failures by means of identifying some specific outages that might possibly lead to catastrophe out of combinatorially large number of possibilities. It presents the formulation of the same as a multi objective optimization problem that partitions its topology graph, accounts for maximizing the imbalance between generation and load in each island and at the same time minimizes the number of lines cut to realize the partitions and employs NSGA II. Experiments were conducted on an IEEE

30 bus test system and the results have been presented. In future, the authors plan to incorporate the same as a separate grid service in their ongoing Grid computing based power system monitoring and control scheme for on line identification of risks in power system.

Acknowledgment. This work has been carried out at the Distributed and Grid Computing Lab, Department of Computer Science and Engineering at National Institute of Science and Technology, Berhampur, 761008. The authors acknowledge the support provided.

References

1. Jian, P., McCalley, J.: On-line analysis of high-order contingencies. In: Proc. North Amer. Power Symp., pp. 297–302 (2001)
2. Ekwue, A.O.: A Review of Automatic Contingency Selection Algorithms for Online Security Analysis. In: The Third International Conference on Power System Monitoring and Control, pp. 152–155 (1991)
3. Phadke, A.G., Thorp, J.S.: Expose Hidden Failures to Prevent Cascading Outages in Power Systems. IEEE Comput. Appl. Power 9(3), 20–23 (1996)
4. Sen, A., Ghosh, P., Vittal, V., Yang, B.: A New Min-Cut Problem with Application to Electric Power Network Partitioning. European Transactions on Electrical Power 19(6), 778–797 (2009)
5. Donde, V., López, V., Lesieutre, B.C., Pinar, A., Yang, C., Meza, J.: Identification of severe multiple contingencies in electric power networks. In: Proceedings of the North American Power Symposium (2005)
6. Lesieutre, B.C., Roy, S., Donde, V., Pinar, A.: Power System Extreme Event Screening using Graph Partitioning. In: IEEE 38th North American Power Symposium, pp. 503–510 (2006)
7. Osyczka, A.: Multicriteria optimization for engineering design. In: Gero, J.S. (ed.) Design Optimization, pp. 193–227. Academic Press, New York (1985)
8. Deb, K.: Multi Objective Optimization Using Evolutionary Algorithms. Wiley, New York (2001)
9. Deb, K., Pratap, A., Agarwal, Meyarivan, S.: A fast and elitist multi-objective genetic algorithm:NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 181–197 (2002)
10. Power System Test Case Archive, `http://www.ee.washington.edu/research/pstca/pf30/pg_tca30bus.html` (last accessed on July 28, 2011)
11. 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)
12. Zhou, A., Qu, B.-Y., Li, H., Zhao, S.-Z., Suganthan, P.N., Zhang, Q.: Multiobjective Evolutionary Algorithms: A Survey of the State-of-the-art. Swarm and Evolutionary Computation 1(1), 32–49 (2011)

Multi Resolution Genetic Programming Approach for Stream Flow Forecasting

Rathinasamy Maheswaran and Rakesh Khosa

Department of Civil Engineering,
Indian Institute of Technology Delhi, Hauz Khas, New Delhi-110016, India
maheswaran27@yahoo.co.in, r.khosa@gmail.com

Abstract. Genetic Programming (GP) is increasingly used as an alternative for Artificial Neural Networks (ANN) in many applications viz. forecasting, classification etc. However, GP models are limited in scope as their application is restricted to stationary systems. This study proposes use of Multi Resolution Genetic Programming (MRGP) based approach as an alternative modelling strategy to treat non-stationaries. The proposed approach is a synthesis of Wavelets based Multi-Resolution Decomposition and Genetic Programming. Wavelet transform is used to decompose the time series at different scales of resolution so that the underlying temporal structures of the original time series become more tractable. Further, Genetic Programming is then applied to capture the underlying process through evolutionary algorithms. In the case study investigated, the MRGP is applied for forecasting one month ahead stream flow in Fraser River, Canada, and its performance compared with the conventional, but scale insensitive, GP model. The results show the MRGP as a promising approach for flow forecasting.

Keywords: Wavelet Analysis, Genetic Programming, Multiscale Forecasting, Stream flow.

1 Introduction

Genetic Programming (GP), as a soft computing technique, has seen successful application in a diverse set of research studies in the recent past ([1], [2], [3]). In the broad area of Civil Engineering, the GP techniques have been applied successfully to develop a rainfall runoff modelling framework by [4], for runoff forecasting by [5], and for optimal structural design by [6]. Further, [7] have used GP to derive a unit hydrograph for an urban basin. GP has also been applied for predicting coastal blooms [8], for detection of incidents on highways [9], for modelling the dynamics of evapotranspiration process [10], and for modelling suspended sediment transport [11]. Other notable GP based modelling studies include flood routing study by [12], daily reference ET estimation by [13], modelling trap efficiency of reservoirs by [14] and forecasting sea level rise by [15]. A consensus view that emerged from these studies was that, overall, GP based models performed better at these tasks in comparison to Artificial Neural Networks (ANN) based models.

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 714–722, 2011.
© Springer-Verlag Berlin Heidelberg 2011

Traditional GP based models, however, present a serious difficulty as these latter models are incapable of accommodating time invariance or non-stationarity. In the real-world context, systems are seldom stationary and application of these techniques to problems involving some degree of non-stationary behaviour in the underlying systems require that the GP based model be updated regularly with newer, or current, information by using recursive techniques like Kalman Filter. The present study adopts an alternative mechanism to handle a non-stationary time series by seeking to couple Genetic Programming with wavelet transforms.

Wavelet transforms is a powerful mathematical technique and has been extensively applied for multi resolution analysis based modelling of numerous types of time series including those that are commonly encountered in the realm of hydrology. Past research has established that the latter, wavelets based, approach has the intrinsic capability to discriminate between various components, time varying or otherwise, of a typical time series and in the process is able to sift out useful information on underlying trends, periodicities and other behavioural attributes that constitutes the given time series. In the recent past, wavelets have been successfully combined with the different soft computing techniques in various modelling and forecasting studies. For example, a conjunctive wavelet and ANN model was developed for drought forecasting [16]. Wavelet networks were proposed for prediction of short and long term hydrological time series [17]. Similarly, [18] proposed a wavelets based Multiscale ARMAX model for time series modelling while a coupled wavelets-ANN model was used to develop a 1 day ahead and 1 month ahead runoff forecasting framework by [19] and [20] respectively. Additionally, [21] and [22] have seen application of these combination approaches for monthly and daily flow forecasting in semi arid watershed.

In some recent studies (see, for example, [12], [14]), GP based models have been shown to outperformed ANN models and this has resulted in a heightened interest in the former approach. Historically too, ANN based models have often been criticized on aspects such as issues of over training, requirement of heuristics in finding the suitable network parameters, and on account of it being a complete black box model. In comparison, on the other hand, GP based approach is gradually emerging as a method of choice as it offers advantages of being an evolutionary algorithm and is able to produce a functional form. Additionally, and importantly, this approach does not require any kind of a priori assumption on model structure to be used for system identification. The purpose of this paper is to investigate the performance of Wavelet based Multi Resolution Genetic Programming (MRGP) model for monthly stream flow forecasting and compare it with the performance of single scale GP and ANN model.

2 Genetic Programming

GP is similar to Genetic Algorithm (GA) but utilizes a "parse tree" structure in its search for solutions, whereas the conventional GA employs bit strips. The technique is a "bottom up" process as it does not require any a priori assumption regarding the structure of the relationship between the independent and dependent variables. The method, on the other hand, identifies an appropriate relationship and may be in the form of logical statements, or a mathematical expression in some familiar mathematical format, or it may even be in the form of an assembly of mathematical functions in a completely unfamiliar format. The GP implementation of relationships has two components: (i) a parse tree, which is a functional set of basic mathematical operators (such as +, -, *, ^, /, sqrt, log, sin, cos), emulating the role of RNA and (ii) the actual components of the functions and their parameters (referred to as the terminal set), which emulate the role of proteins or chromosomes in biological systems. These two components work simultaneously to mimic the dynamics of evolutionary processes. A detailed discussion on theoretical development of the framework for GP based approach is beyond the scope of this paper and is extensively discussed in literature (see, for example, reference cited at [1] for a detailed discussion).

3 Maximal Overlap Discrete Wavelet Transform (MODWT) and Multi Resolution Analysis

The MODWT is a mathematical tool that projects a time series onto a collection of non-orthogonal basis functions (wavelets) to produce a set of wavelet coefficients [23]. These coefficients capture information from the time series at different frequencies and at distinct instants. For a redundant transform like the MODWT, an *n* samples input time series will have *n* samples resolution scale for each resolution level. Therefore, the features of wavelet coefficients in a Multi Resolution Analysis (MRA) will be lined up with the original time series in a meaningful way.

For a time series *X* with n samples, the MODWT yields an additive decomposition or MRA given by,

$$X = \sum_{j=1}^{N} D_j + S_N \tag{1}$$

According to Equation (1), a set of coefficients $\{D_j\}$, corresponding to respective scale *j*, can be obtained with the same number of samples (*n*) as in the original signal (*X*). These are called wavelet “details” and they capture local fluctuations over the whole period of a time series at that particular scale. The set of values S_N provide a “smooth” or overall “trend” of the original signal. Adding D_j to S_N, for $j = 1, 2... N$, (where N is the number of decomposition level) gives an increasingly more accurate approximation of the original signal.

4 Model Development

MRGP model is obtained by combining the two methods, GP and Wavelet Transform. As a first step, scale specific decompositions of the given time series are derived using wavelet transform and is followed with the development of a scale specific GP model for its respective scale specific decomposition level. The individual contributions obtained thus for each scale are then combined linearly to obtain forecast of the modelled state variable at next time horizon. From equation (1), any discrete time series $X = (X_k)_{k=1,2,\ldots n}$ can be written in the following form

$$X_k = S_{N,k} + \textstyle\sum_{j=1}^{N} D_{j,k} \qquad k = 1,2, \ldots .n \tag{2}$$

To compute the value of $\hat{X}_{k+1}$, the one step ahead forecast value of X_{k+1}, and made at time step k, it is sufficient to calculate
$\hat{S}_{k+1}$ and $\hat{D}_{j,k+1}\, j = 1,2, \ldots .N$.

Using the GP evolved functions we can estimate $\hat{S}_{k+1}$ and $\hat{D}_{j,k+1}$ using the model having a general form,

$$\hat{S}_{N,k+1} = \hat{f}_0(S_{N,k}, S_{N,k-1}, \ldots .S_{N,k-q,}) \tag{3}$$

$$\hat{D}_{j,k+1} = \hat{f}_j(D_{j.k}, D_{j.k-1}, \ldots \ldots .D_{j.k-q,}), j = 1,2, \ldots \ldots N \tag{4}$$

The functional forms of f_o and f_j are obtained by minimizing the sum of squared errors and q is order (or memory) of the model at each scale. Finally, the integration of scale wise forecasts at each level is linearly combined to yield the forecast of the model state variable $\hat{X}_{k+1}$ using the expression:

$$\hat{X}_{k+1} = \hat{S}_{N,k+1} + \sum_{j=1}^{N} \hat{D}_{j,k+1} \tag{5}$$

The entire model scheme is shown in Figure 1.

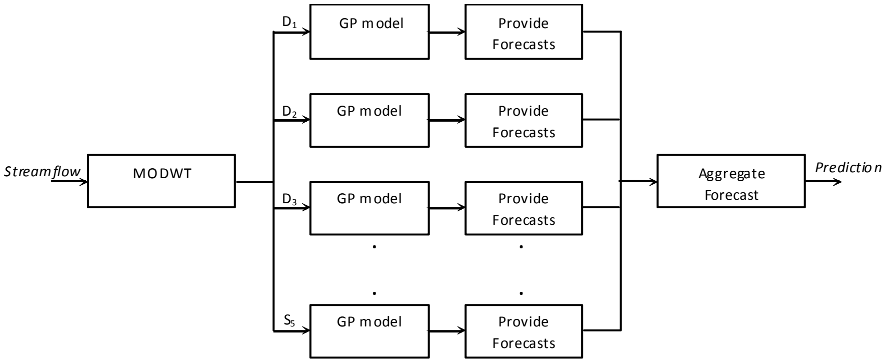


Fig. 1. MRGP MODEL for stream flow forecasting

The performance of the proposed approach is compared with the single scale traditional GP, ANN and ARIMA models in terms of the residual error analysis and other performance indices such as Root Mean Square Error (RMSE), Coefficient of determination (R^2), and Mean Absolute Error (MAE). The RMSE, R^2 and MAE are defined as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{M}(Q_i - \hat{Q}_i)^2}{M}} \tag{6}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{M}(Q_i - \hat{Q}_i)^2}{\sum_{i=1}^{M}(Q_i - \bar{Q}_i)^2} \tag{7}$$

$$MAE = \frac{1}{M}\sum_{i=1}^{M}\left|Q_i - \hat{Q}_i\right| \tag{8}$$

In the equations given above, M is the number of data sets or observations available for use, Q_i is the observed value and $\hat{Q}_i$ is the model predicted value.

5 Model Application

In order to investigate the efficiency of the proposed method, the MRGP was applied to develop 1-month ahead monthly flow forecasts for Fraser River in Canada. In the study reported by [24], this time series was investigated and the authors indicated presence of non-stationarity in terms of multi-scale hidden dynamics. Figure 2 presents a plot of these mean monthly flows from January, 1913 to December, 1990. For objectivity, models were designed based on the split sample test approach in which data for the first 50-years were used for calibration, the next 20-year reserved for testing and the remaining 10 years of flow data was used for validating the model. As a first step, the 'db2' wavelets transform approach was used to obtain 5 level decompositions of the given stream flow time series. Independent GP models, such as the one given as equation 13, were designed for each scale and following a comparative evaluation, the best one was adopted as the model of choice for further analysis. As a next step, the model outputs were then combined to forecast 1- month ahead stream flow. Table 1 shows the general GP parameters used in this study. Different input combinations were tested for selecting the best ANN and GP models. Table 2 shows the models' performance statistics (R^2, RMSE and MAE) for the best GP and MRGP model, ANN and the ARIMA model. The analyses of the results clearly show the improved performance of the MRGP model when compared with the single scale GP model.

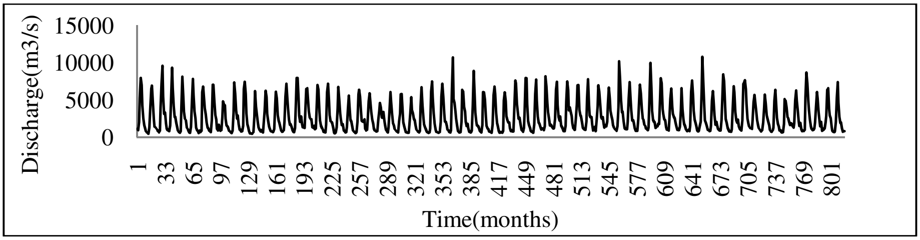


Fig. 2. Average Monthly Streamflow of Fraser River

Table 1. GP Parameters

S.No	Parameter	Fraser river
1	Population Size	300
2	Mutation Frequency (%)	95
3	Crossover Frequency (%)	50
4	Program Size	
	Intial	80
	Maximum	512
5	Total Runs	15
6	Cost Function	MSE
7	Set Functions	' +,-,x,/,sqrt(),sin()'
8	Inputs to GP Model	$X_{t-1}, X_{t-2}, X_{t-3}, X_{t-11} X_{t-12}$

Table 2. Model Performance for the Fraser river series – One month ahead prediction

		Testing			Validation		
S.no	Model Type	RMSE(m3/s)	R^2	MAE(m3/s)	RMSE(m3/s)	R^2	MAE(m3/s)
(i)	GP	761.23	0.86	525.27	799.55	0.825	554.97
(ii)	MRGP	683.32	0.92	480.23	640.23	0.89	494.73
(iii)	ARIMA	804.3	0.76	570.32	845.32	0.72	600.35
(iv)	ANN	780.25	0.83	540.45	820.47	0.78	577.95

From Table 2 and Figures 3 and 4, it can be seen that the MRGP model outperforms all the other competing models for the desired 1-month ahead forecasts. It is seen that the MRGP model has a validation period R^2 and RMSE values equal to 0.89 and 640.23m^3/s respectively. In comparison, the best GP and ANN models had a validation period R^2 of 0.825 and 0.78 respectively while the reported RMSE values were equal to 799.5 m^3/s and 820.47m^3/s respectively for these models. The evolved, final, functional form for the GP model was accordingly obtained as:

$$Q_{t+1} = \frac{22.12(Q_{t-11} - Q_{t-12})^2}{Q^2_{t-9}} + 0.159\left[\frac{Q_{t-12}Q_{t-11} - Q^2_{t-12}}{Q_{t-9}}\right] + 0.46Q_{t-1} + 0.69Q_{t-12} - 0.159Q_{t-2} \tag{9}$$

Equation (13) is the simplified form of the best model evolved by the GP. The equation presents a complex and nonlinear model for one month ahead forecast.

Figures 3 and 4 shows the comparison of performances of the GP and MRGP derived estimates respectively with the corresponding observations of the modelled flow state variable. These figures clearly establish the superior performance of the wavelets based approach as can be seen in terms of an improved description of peak flow rates as well as their time of occurrence.

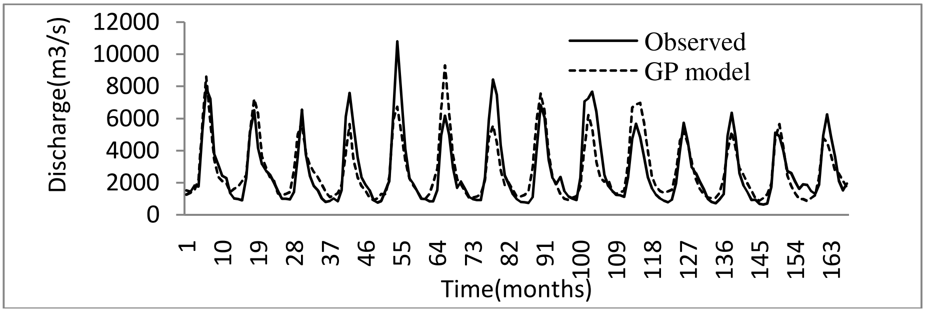


Fig. 3. Model Performance for the Single Resolution GP for Fraser River (Validation Results)

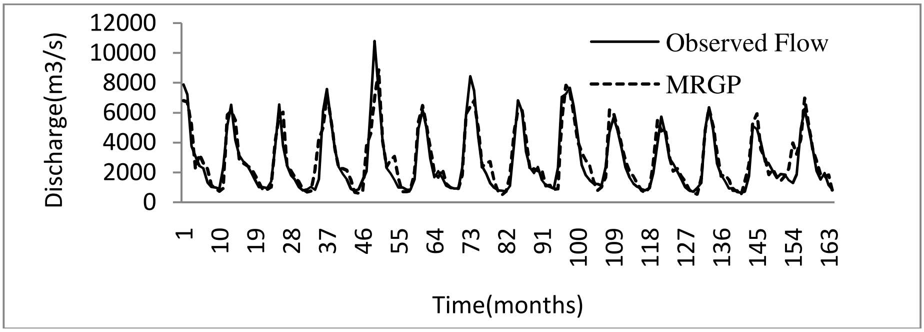


Fig. 4. Model Performance for the MRGP for Fraser River (Validation Results)

6 Conclusions

The potential of Multi Resolution Genetic Programming (MRGP) model for streamflow forecasting was investigated in this study for a non-stationary time series. The MRGP models were developed by combining two methods, namely the Wavelet Transform and Genetic Programming. For the investigated case study of monthly stream flows, the proposed methodology has significantly improved the forecast accuracy, as indicated by the associated values of RMSE, by as much as 20% and 22% respectively when compared with the single resolution GP and ANN models. Similarly, the improved MRGP based model performance is also indicated by the higher values obtained for the performance indicator R^2. Clearly, MRGP models are more accurate since wavelet transforms provide a better discrimination of non-stationarities that are present at various scales in the given time series and, additionally, provides their corresponding stationary decompositions. This improves the ability of the coupled GP based forecasting model by capturing useful information on various resolution levels. Further, the functional form provided by the MRGP is quite simple and, at the same time, a versatile model and can be directly used for future predictions without involving excessive computation.

References

1. Koza, J., Poli, R.: Genetic programming. Search Methodologies, 127–164 (1992)
2. Banzhaf, W., et al.: Genetic Programming- An introduction (On the automatic evolution of computer programs and its applications). Morgan Kaufmann Publishers, ISBN 1-55860-510-X (1998)
3. Spector, L., Barnum, H., Bernstein, H.J., Swamy, N.: Quantum computing applications of genetic programming. In: Spector, L., Langdon, W.B., O'Reilly, U., Angeline, P.J. (eds.) Advances in Genetic Programming, vol. 3, p. 135. MIT Press, Cambridge (1999)
4. Savic, D., Walters, G., Davidson, J.W.: A genetic programming approach to rainfall-runoff modelling. Water Resources Management 13(3), 219–231 (1999)
5. Khu, S., Liong, S.: Genetic programming and its application in real-time runoff forecasting1. Journal of the American Water Resources Association 37(2), 439–451 (2007)
6. Yang, Y., Soh, C.K.: Automated optimum design of structures using genetic programming. Computers and Structures 80, 1537–1546 (2002)
7. Dorado, J., Rabuñal, J.R., Puertas, J., Santos, A., Rivero, D.: Prediction and Modelling of the Flow of a Typical Urban Basin through Genetic Programming. In: Cagnoni, S., Gottlieb, J., Hart, E., Middendorf, M., Raidl, G.R. (eds.) EvoWorkshops 2002. LNCS, vol. 2279, pp. 190–201. Springer, Heidelberg (2002)
8. Muttil, N., Lee, J.: Genetic programming for analysis and real-time prediction of coastal algal blooms. Ecological Modelling 189(3-4), 363–376 (2005)
9. Roberts, S.C., Howard, D.: Detection of Incidents on Motorways in Low Flow High Speed Conditions by Genetic Programming. In: Cagnoni, S., Gottlieb, J., Hart, E., Middendorf, M., Raidl, G.R. (eds.) EvoWorkshops 2002. LNCS, vol. 2279, pp. 245–254. Springer, Heidelberg (2002)
10. Parasuraman, K., Elshorbagy, A., Carey, S.K.: Modelling the dynamics of the evapotranspiration process using genetic programming. Hydrological Sciences Journal 52(3), 563–578 (2007)
11. Aytek, A., Kisi, Ö.: A genetic programming approach to suspended sediment modelling. Journal of Hydrology 351(3-4), 288–298 (2008)
12. Sivapragasam, C., Maheswaran, R., Venkatesh, V.: Genetic programming approach for flood routing in natural channels. Hydrological Processes 22(5), 623–628 (2007)
13. Guven, A., Aytek, A.: New Approach for Stage–Discharge Relationship: Gene-Expression Programming. Journal of Hydrologic Engineering 14, 812 (2009)
14. Garg, V., Prakash, J.: Reservoir Sedimentation Estimation Using Genetic Programming Technique. World Environmental and Water Resources Congress (2009)
15. Ali Ghorbani, M., Khatibi, R., Aytek, A., Makarynskyy, O., Shiri, J.: Sea water level forecasting using genetic programming and comparing the performance with Artificial Neural Networks. Computers & Geosciences 36(5), 620–627 (2010)
16. Kim, T.: Nonlinear model for drought forecasting based on a conjunction of wavelet transforms and neural networks. Journal of Hydrologic Engineering 8, 319 (2003)
17. Wang, W., Ding, J.: Wavelet network model and its application to the prediction of hydrology. Science 1(1), 67–71 (2003)
18. Chen, H., Vidakovic, B., et al.: Multiscale forecasting method using armax models. Technological Forecasting and Social Change (2004)
19. Anctil, F., Tape: An exploration of artificial neural network rainfall-runoff forecasting combined with wavelet decomposition. J. Environ. Eng. Sci. 3(S1), S121–S128 (2004)

20. Cannas, B., Fanni, A., See, L., Sias, G.: Data preprocessing for river flow forecasting using neural networks: Wavelet transforms and data partioning. Physics and Chemistry of the Earth 31(18), 1164–1171 (2006)
21. Kisi, O.: Stream flow forecasting using neuro-wavelet technique. Hydrological Processes 22(20), 4142–4152 (2008)
22. Adamowski, J., Sun, K.: Development of a coupled wavelet transform and neural network method for flow forecasting of non-perennial rivers in semi-arid watersheds. Journal of Hydrology 390(1-2), 85–91 (2010)
23. Percival, D.B., Walden, A.T.: Wavelet methods for time series analysis. Cambridge Univ. Press (2006)
24. Ferreira, M., West, M., et al.: Multi-scale and hidden resolution time series models. Bayesian Analysis 1(4), 947–968 (2006)

Reference Set Metrics for Multi-Objective Algorithms

Chilukuri K. Mohan and Kishan G. Mehrotra

Department of Electrical Engineering & Computer Science,
Syracuse University, Syracuse, NY, USA
`{ckmohan,mehrotra}@syr.edu`

Abstract. Several metrics and indicators have been suggested in the past to evaluate multi-objective evolutionary and non-evolutionary algorithms. However, these metrics are known to have many problems that make their application sometimes unsound, and sometimes infeasible. This paper proposes a new approach, in which metrics are parameterized with respect to a reference set, on which depend the properties of any metric.

Keywords: Multi-objective Algorithms, Reference Set, Metrics.

1 Introduction

Evaluating and comparing single-objective evolutionary algorithms is a relatively straightforward task: *evaluate whether or not a particular solution quality was achieved, how often (over various trials) such quality was achieved, and how much computational effort was required.* By contrast, the evaluation of multi-objective evolutionary and non-evolutionary algorithms, collectively abbreviated as MOAs in this paper, is rendered difficult by the lack of simple and satisfactory performance metrics[1]. The main reason is that the output of a MOA run is a collection of vectors forming a non-dominated set. Comparative results are generally shown in graphical form indicating which algorithm performs better [7,8]. Some oft-used metrics are discussed in [6,10]; a comprehensive list of metrics is available in Table 1 on page 41 of a recent survey article by Zhou et. al [9]. For example, several of the classical metrics are defined with respect to the Pareto set consisting of all non-dominated solutions to the problem. Unfortunately, the Pareto set is unknown or impossible to enumerate in many practical problems. Other metrics rely on computing the hyper-volume dominated by solutions produced by an algorithm; an algorithm that dominates greater volume is considered to be better than another that dominates less volume.

The unsoundness or unsatisfactoriness of several oft-used metrics has been discussed by Knowles and Corne [6]. Some difficulties are: potential non-transitivity[2],

---

[1] The word “indicator” better describes these, but MOA literature appears to prefer the slightly inaccurate usage –“metric”; used in this paper as well.

[2] It is possible that algorithm A is considered better than algorithm B, and B better than C, as well as C better than A are fundamental to any 3-party election.

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 723–730, 2011.
© Springer-Verlag Berlin Heidelberg 2011

consequences of Arrows theorem [1], and Condorcets voting paradox [12], resolutions to which have been discussed by researchers in Economics and Game Theory.

In this paper we propose that all comparative evaluations of MOAs should be with respect to a Reference Set (RefSet). Inferences and conclusions based on such evaluations cease to be valid when the context of the RefSet is removed, yet in some instances, this is the only possible approach. Various metrics can be proposed, similar to those existing in the literature, but parameterized by a specific RefSet. The properties satisfied by a metric would then depend on the choice of the RefSet. Valid statistical arguments can be constructed to evaluate algorithms in case one algorithm is shown to be better than other with many unbiased choices of RefSets. Depending on the choice of RefSets, it may be possible to combine the results with appropriate weights to obtain an *overall* performance measure.

Section 2 presents the main idea of this paper, discussing why and how RefSets may be constructed to facilitate algorithm comparison. Sections 3-5 focus on RefSet-based metrics related to solution set cardinality, domination area, and solution set diversity. Predictably, the last section presents concluding remarks.

2 Reference Sets

A Reference Set (RefSet) is a collection of candidate solutions with respect to which we can compare two algorithms. We may distinguish between RefSets that:

- Focus exclusively on non-domination, abbreviated NRefSets;
- Focus exclusively on diversity, abbreviated DRefSets; and
- Consider both non-domination and diversity, abbreviated DNRefSets.

For example, a possible NRefSet that can be used by the metric is the Pareto set consisting of all non-dominated solutions to the problem. Unfortunately, as mentioned earlier, the Pareto set is unknown or impossible to enumerate in many practical problems. Such metrics can instead be replaced by others that are parameterized with an appropriately chosen RefSet. The following are examples of RefSets that can be used to compare algorithms with respect to a specific problem instance:

- U: the union of all solutions to the problem instance obtained by all means known to humanity for some benchmark problems, the Pareto set may be available for use as U.
- NU: the subset of U consisting only of mutually non-dominating solutions, i.e., obtained by deleting all elements of U that are dominated by other elements in U.
- U_D : a subset of U obtained by deleting elements of U that are near others according to a minimal distance threshold condition D specified to hold between any two elements in the set. The threshold conditions may be

  - In data space,
  - In objective function space.

  The latter is desirable from the perspective of sampling different parts of the Pareto surface, whereas the former is of interest in applications such as routing where robustness is to be achieved by finding multiple substantially distinct solutions. The distance threshold condition D may be parameterized based on the threshold value used to evaluate proximity. Also, different choices of U_D can be obtained for the same $U$and the same condition D, since the choice of the elements being deleted may be arbitrary.
- NU_D : obtained by applying the distance threshold restriction D to NU.
- $Uk, NUk, Uk_D, NUk_D,$: similar to above; consisting of the union of solutions to the problem instance obtained by k algorithms being compared, where $k \geq 2$.
- $U-, NU-, U_{-D}, NU_{-D}$: similar to above, beginning with the union of solutions to the problem instance obtained by a collection of algorithms **excluding** the one being evaluated.

The choice of the reference set depends on the properties considered to be of importance for a specific algorithm comparison. Reference sets formulated without utilizing the solutions obtained using a specific algorithm are ideal, providing objective criteria that do not depend on the vagaries of the algorithms being evaluated; all the sets listed above satisfy this condition, except Uk, NUk, Uk_D, NUk_D, As a reasonable and sound experimental methodology, for instance, when two specific algorithms are being evaluated against each other, all **other** available algorithms may be used to generate the reference set.

Example: When

(a) the requirements of considering both non-domination and solution set diversity hold,
(b) only two algorithms are being compared,
(c) the Pareto set is unknown, and
(d) no other algorithms have so far been applied to that problem instance,

the right choice of the reference metric is expected to be $NU2_D$, where D eliminates some candidate solutions whose distance to others in the set (in objective space) is less than a prespecified problem-specific threshold. For instance, for the automobile buyer's decision-making problem with twin objectives of cost and comfort, this approach may delete one of two elements whose cost differs by less than \$200 and estimated comfort level differs by less than 0.1. Alternatively, D may be specified in data space instead of objective space, e.g., $NU2_D$ may be prohibited from containing two cars made by the same manufacturer. Note, as in these examples, that the distance-related criterion D need not depend on Euclidean distance in multi-dimensional space. Also, the choice of D does not uniquely determine $NU2_D$, in general.

The high-level approach we propose introduces the RefSet parameter into existing metrics. The next three sections explore the approach, developing RefSet-based metrics that are similar to several kinds of existing metrics.

3 Solution Set Cardinality

Counting the number of non-dominated solutions produced by an algorithm is a procedure that has come under some criticism, e.g., [6]. An algorithm could be rated highly because it produces a large number of candidate solutions, even when compared to another algorithm that produces a single Pareto-optimal solution that dominates all the others.

However, set cardinality can make more sense with respect to a RefSet, obtaining ordinal or comparative measures such as the following:

- An algorithm can be evaluated using the fraction of elements in the RefSet that are contributed by another algorithm; an algorithm can be considered to be better than another, with respect to the RefSet, if it contributes more elements to the RefSet.
- A milder, more practical criterion is to count the fraction of elements in the RefSet that are very near (according to some reasonable distance threshold in the objective space) to candidate solutions produced by an algorithm.

These measures do not directly rely on the cardinality of the solution set produced by the algorithms, but instead depend on the relationship between the RefSet and the solutions produced by the algorithms. The following properties hold for the above criteria:

- Algorithms are not rewarded for producing multiple near-identical candidate solutions.
- Algorithms are not directly penalized for producing only a small number of very good solutions.

Example: A RefSet consisting of $NU2$, augmented by problem-specific knowledge, contains three solutions from algorithm A, two solutions from algorithm B, and two others that are almost identical in objective function values to other solutions from B. Then, the first comparative measure argues that A is better than B since $3 \geq 2$, whereas the second measure supports the opposite conclusion, since B has generated solutions that are very near to four solutions in the RefSet.

Properties: We consider here some of the properties satisfied by a specific RefSet-based metric, suggesting the usefulness of this methodology. Let the ordinal metric $F(X;R)$ be defined as the fraction of solutions in RefSet R obtained using algorithm X, and the associated comparison metric $B_F(X,Y;R)$ according to which algorithm X is better than algorithm Y if $F(X;R) \geq F(Y;R)$. This metric satisfies the following properties, in some cases only for appropriate choices of R:

- Transitivity: $(B_F(X,Y;R)$ and $B_F(Y,Z;R))$ implies $B_F(X,Z;R)$.
- Antisymmetry: $B_F(X,Y;R)$. implies $\neg B_F(Y,X;R)$.

- Compatibility with outperformance relations[3],[4]: Let all points in B be equal to or dominated by points in A, and let A contain at least one point that is not in B. Then $BF(A, B; NU2)$, but not necessarily $BF(A, B; NU)$ nor $BF(A, B; NU-)$ since none of the solutions in NU (or $NU-$) may come from A. In other words, an algorithm may produce solutions that dominate the solutions of another algorithm. But it is possible that both algorithms are equally bad as far as the RefSet is concerned, especially when the RefSet is constructed independently of the algorithms being compared.

Such analysis can be carried out for various metrics, identifying RefSets for which various properties such as [weak] compatibility with weak/strong/complete outperformance hold.

4 Domination Volume

Problems with the metric that uses domination volume involve the choice of the reference point (the origin in the graph if all objectives are to be maximized), as well as the incommensurability of different objective function dimensions[5]. Changing the units of measure from feet to meters or pounds to kilograms can result in substantially different results with respect to volume-comparison measures. The first decision, concerning the choice of the origin or reference point, involves asking the following question: which parts of the objective function space are truly of interest to the practitioner? For example, the automobile purchase decision-maker may rule out all vehicles costing more than $60,000, as well as those that do not meet minimal comfort standards. The second obstacle to volume-comparison measures, viz., non-commensurability of objective function dimensions, can be addressed to some extent by using a non-uniform problem-dependent scaling of the axes (objective functions) prior to the volume computation.

For instance, the decision-maker can be queried to determine the interval of values between which he/she is relatively indifferent, at various possible values for each objective function.

Example: An automobile purchaser may consider that a difference of $500 is significant when the purchase price is about $10,000, but not significant when the purchase price is about $60,000. Using this strategy, the two-dimensional objective function space could be divided into a collection of rectangular cells using non-uniformly spaced grid lines, and the volume comparison metric is transformed into one of comparing which cells are dominated by solutions generated by one algorithms but not the other. This approach provides a coarse measure

---

[3] Knowles and Corne [6] define the concepts of 'outperformance' and 'compatibility'.

[4] For ease of reading, we abuse notation, identifying algorithms with the nondominated solution sets they produce; Greek letters can be invoked if needed.

[5] Apples and oranges can be multiplied, but products of differences in the numbers of apples and oranges cannot be compared!

for algorithm comparison that is more robust than the volume comparison metric, although it remains subject to the criticism that a cell in one part of the objective function space cannot be considered equivalent in size or importance to a cell in a completely different part of the space.

For example, twenty cells are dominated by one algorithm but not by the other, whereas forty cells are dominated by the second algorithm but not the first. Superficially, the second algorithm is better using the cell-counting metric, but the justifiability of this comparison depends on the appropriateness of the problem-specific choice of the grid lines (or planes or hyperplanes) used to separate the objective function space into cells.

By moving from objective function dimension-based volume measures to more abstract cell-counting measures, the above discussion provides a guideline for the choice of volume-like RefSet-based metrics. These metrics are constructed using elements in the RefSet to determine cells, using an approach such as Voronoi tessellation in which each element of the RefSet corresponds to one cell or region in objective function space.

The applicability of this approach relies to some extent on the degree to which elements of the RefSet satisfactorily describe the Pareto set. When the true Pareto set is unavailable, of course, such reasoning is based on the assumption that the RefSet is the best available approximation to the Pareto set. Further, there is an implicit assumption that two cells are roughly equivalent in importance, suggesting that the RefSet should be modified or thinned out (e.g., NU_D defined in Section 2) to eliminate elements that are near-identical in objective function space.

5 Diversity

Classic Pareto set coverage and related metrics address how well the solutions generated by an algorithm encompass the Pareto-optimal solutions appropriate to the problem instance being considered. Where the Pareto set is known or easily determinable, this is exactly as the special case where the RefSet is the same as the Pareto set.

In practical situations, we would instead have to use the RefSets described in Section 2, instead of the true Pareto set. The coverage issue is partly addressed by the metrics mentioned in Section 3 and 4, but the general question still remains: how do we evaluate an algorithm that generates some solutions that belong to the RefSet, and others that are near but are not identical to other solutions in the RefSet?

One aspect of this problem involves checking whether the solutions obtained by an algorithm are spread out widely in the objective function space, with respect to elements in the RefSet. If two algorithms generate equal numbers of elements in the RefSet, but one generates elements that are clustered together in objective function whereas the other does not, we may argue that the second is better than the first. One metric, to which this discussion leads, involves computing the average distance from each RefSet element to the nearest candidate solution generated by an algorithm, averaging over all RefSet elements but not averaging over all the solutions produced by the algorithm.

A related but different issue is the internal diversity of the solution set obtained by an algorithm (without reference to a RefSet). How widely spaced apart are elements of a solution set? However, spacing measures that focus exclusively on this issue are not very desirable; in particular, an algorithm that generates a vast number of well-spaced candidate solutions is not necessarily a good one, since many of these solutions may be of relatively poor quality with respect to the domination criterion, e.g., they may all be dominated by a single solution generated by a different algorithm. Such diversity metrics make sense only when considered along with a specific RefSet, as in the metric mentioned in the preceding paragraph.

6 Discussion

This paper has proposed that multi-objective optimization algorithms be evaluated using metrics that depend on Reference Sets. The goals of the comparison and problem details would be relevant in choosing the appropriate reference set. Some reasonable choices of reference sets were listed in Section 2. We have also discussed how we may adapt the metrics previously proposed in the literature to evaluate the number, diversity, and coverage properties of algorithms. However, we have only provided a framework for the formulation of such metrics in this paper. A careful study of the properties of a specific metric, in the lines of the evaluative study of [6], is needed before applying it to evaluate a specific algorithm. The last paragraph at the end of Section 3 shows that the properties of the metric depend significantly on the choice of the RefSet.

Acknowledgments. The authors thank Ramesh Rajagopalan and Pramod Varshney for prior discussions that helped formulate the ideas in this paper.

References

1. Arrow, Social Choice and Individual Values, 2nd edn. John Wiley & Sons, Inc., New York (1963)
2. Coello Coello, C.A.: A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniqes. Knowledge and Information Systems: An International Journal 1(3), 269–3087 (1999)
3. Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons, Chichester (2001)
4. Fonseca, C.M., Fleming, P.J.: An overview of evolutionary algorithms in multiobjective optimization. Evolutionary Computation 3(1) (Spring 1995)
5. Knowles, J.D.: Local-Search and Hybrid Evolutionary Algorithms for Pareto Optimization. The University of Reading, Reading (2002)
6. Knowles, J.D., Corne, D.: On Metrics for Comparing Nondominated Sets. In: Congress on Evolutionary Computation (CEC 2002), vol. 1, pp. 711–716. IEEE Service Center, Piscataway (2002)

7. Rajagopalan, R., Mohan, C.K., Mehrotra, K.G., Varshney, P.K.: EMOCA: An Evolutionary Multi-Objective Crowding Algorithm. Journal of Intelligent Systems 17(1-3), 107–123 (2008)
8. van Veldhuizen, D.A., Lamont, G.B.: Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-art. Evolutionary Computation 8(2), 125–147 (2000)
9. Zhou, A., Qu, B.-Y., Li, H., Zhao, S.-Z., Suganthan, P.N., Zhang, Q.: Multiobjective evolutionary algorithms: A survey of the state-of-the-art. Swarm and Evolutionary Computation 1(1), 32–49 (2011)
10. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., da Fonseca, V.G.: Performance Assessment of Multiopjective Optimizers: An Analysis and Review. IEEE Transactions on Evolutionaryy Computation 7(2), 117–130 (2003)
11. Zhou, A., Qu, B.-Y., Li, H., Zhao, S.-Z., Suganthan, P.N., Zhang, Q.: Multiobjective evolutionary algorithms: A survey of the state-of-the-art. Swarm and Evolutionary Computation 1(1), 32–49 (2011)
12. A wikipedia document discussing Condorcets Voting Paradox, `http://en.wikipedia.org/wiki/Votingparadox`

Groundwater Level Forecasting Using SVM-QPSO

Ch. Sudheer[1], Nitin Anand Shrivastava[2], Bijaya Ketan Panigrahi[2], and Shashi Mathur[1]

[1] Department of Civil Engineering,
Indian Institute of Technology, Hauzkhas, New Delhi, India
[2] Department of Electrical Engineering,
Indian Institute of Technology, Hauzkhas, New Delhi, India

Abstract. Forecasting the groundwater levels in a water basin plays a significant role in the the management of groundwater resources. In this study, Support Vector Machines (SVM) is used to construct a ground water level forecasting system. Further Quantum behaved Particle Swarm Optimization function is adapted in this study to determine the SVM parameters. Later, the proposed SVM-QPSO model is employed in estimating the groundwater level of Rentachintala region of Andhra Pradesh in India. The performance of the SVM-QPSO model is then compared with the ANN (Artificial Neural Networks). The results indicate that SVM-QPSO is a far better technique for predicting groundwater levels as it provides a high degree of accuracy and reliability.

1 Introduction

The main aim of this study is to tune parameters of SVM using PSO to enhance the performance of the SVM models, which have gained immense growing popularity of late [1,2,3,4,5,6,7,8,9,10]. SVM application studies are performed by 'expert' users and Since the quality of SVM models depend on a proper setting of SVM hyper-parameters, the main concern for researchers is to set these parameter values (to ensure good generalization performance) for a given data set in SVM regression. While the existing study on SVM regression [10,11,12] provide some recommendations for appropriate setting of SVM parameters, there is hardly general consensus and many contradictory opinions regarding these settings. Hence, re-sampling seems to be the possible method of choice for many applications. Unfortunately, using re-sampling for tuning several SVM parameters is very expensive in terms of computational costs and data requirements. So the basic objective of this study thus is to determine ways and means for obtaining the optimal parameter values of SVM. In this study Qunatum behaved Particle swarm optimization (PSO) technique is employed to find out SVM parameters.

PSO was introduced by Eberhart and Kennedy [13], inspired by the social behavior of animals such as bird flocking, fish schooling, and the swarm theory. Compared with GA and other similar evolutionary techniques, PSO has some attractive characteristics and in many a cases proved to be more effective [14]. Both GA and PSO have been used extensively for a variety of optimization problems and in most of these cases PSO has proven to have superior computational efficiency [14]. Since 1995, many attempts have been made to improve the performance of the PSO [15,16]. Sun and Xu [17] introduced quantum theory into PSO and propose a Quantum-behaved PSO (QPSO)

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 731–741, 2011.
© Springer-Verlag Berlin Heidelberg 2011

algorithm, which is guaranteed theoretically to find good optimal solutions in search space. The experiment results on some widely used benchmark functions show that the QPSO works better than standard PSO [17] and is a promising algorithm. In this study Quantum-behaved Particle swarm optimization (QPSO) technique is employed to find out SVM parameters. This paper is organized as follows. Section 2 gives a brief introduction to SVM algorithm. Section 3 describes the adaptation of QPSO algorithm in estimating the optimal SVM parameters. Section 4 gives the brief description about the study area, the available data and the model construction. In Section 5 the performance of SVM-PSO model is compared with ANN (Artificial Neural Networks). Finally, the conclusions are presented in Section 6.

2 Support Vector Machines

In this section, the theory behind SVM equations is given, based on Vapnik's formulation [18]. Assume $\{(x_1,y_1),....,(x_\ell,y_\ell)\}$ be the given training data sets, where each $x_i \subset R^n$ shows the input space of the sample and has a corresponding target value $y_i \subset R$ for $i = 1,\dots,l$ Where l represents the size of the training data . The basic principle of regression problem is to find a function which has at most ε deviation from the actually obtained targets y_i for all the training data, and at the same time is as flat as possible. In other words, errors are tolerable as long as they are less than ε, but any further deviations greater than ε are not desirable. The equation 1 is estimated in SVM for linear regression.

$$\text{minimize} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=0}^{\ell}(\xi + \xi^*) \tag{1}$$

$$\text{subjected to} \begin{cases} y_i - \langle w,x_i\rangle - b \le \varepsilon_i + \xi \\ \langle w,x_i\rangle + b - yi \le \varepsilon_i + \xi^* \\ \xi_i,\xi_i^* \ge 0 \quad i = 1,\dots,l \end{cases}$$

The $C > 0$ is a regularization constant which determines the trade off between the flatness of f and the tolerated deviations greater than the ε from the loss function. This is achieved by dealing with a so called ε-insensitive loss function as described in equation 2

$$|\xi|_\varepsilon = \begin{cases} 0 & \text{if } |\xi| \le \varepsilon \\ |\xi| - \varepsilon & \text{otherwise} \end{cases} \tag{2}$$

This defines an ε tube so that if the predicted value is within the tube the loss is zero, while if the predicted point is outside the tube, the loss is the magnitude of the difference between the predicted value and the radius ε of the tube. The slack variables, ξ,ξ^* correspond to the size of this excess deviation for upper and lower deviations, respectively.

To solve equation 1, some Lagrangian multipliers $(\alpha_i,\alpha_i^*,\eta,\eta^*)$ are introduced in order to eliminate some of the primal variables. Hence, the Lagrangian of Equation 1 is given as:

$$L_p = C\sum_{i=1}^{\ell}(\xi_i+\xi_i^*)+\frac{1}{2}\|w\|^2-\sum_{i}^{l}(\eta_i\xi_i+\eta_i^*\xi_i^*)-\sum_{i=1}^{l}\alpha_i(-y_i+\langle w,x_i\rangle+b+\varepsilon+\xi_i) \\ -\sum_{i=1}^{l}\alpha_i^*(+y_i-\langle w,x_i\rangle-b+\varepsilon+\xi_i^*) \quad (3)$$

$$\text{subjected to} \quad \alpha_i,\alpha_i^*,\eta,\eta^* \geq 0 \quad i=1,\ldots,l$$

Here L is the lagrangian function and $\alpha_i,\alpha_i^*,\eta,\eta^*$ are Lagarange mutipliers. The Equation 3 is an primal equation it is converted in to dual. As the dual equation ensures a unique global optimum and are easy to solve. Further by some mathematical manipulations with lagarange multipliers the Equation 3 is reduced to equation 4.

$$w=\sum_{i=1}^{l}(\alpha_i-\alpha_i^*)x_i \quad (4)$$

By substituting equation 4 into equation 1, the generic equation can be rewritten as:

$$f(x)=\sum_{i=1}^{l}(\alpha_i-\alpha_i^*)\langle x_i,x\rangle+b \quad (5)$$

The above mentioned approach discussed the mechanism of linear regression. However, nonlinear regression can be performed by replacing x_i with a mapping into the feature space $\phi(i)$ which linearizes the relationship between x_i and y_i.In the feature space, the original approach can be adopted in finding the regression solution. When using a mapping function, the solution of equation 5 becomes:

$$f(x)=\sum_{i=1}^{l}(\alpha_i-\alpha_i^*)K(x_i,x)+b \quad (6)$$
$$\text{where} \quad K(x_i,x)=\langle\phi(x_i),\phi(x)\rangle$$

The function $k(x_i,x_j)$ is known as the kernel function. Kernel functions enable dot product to be performed in high dimensional feature space using low dimensional space data input. The Kernel functions makes it possible to construct linear decision surfaces in the feature space instead of constructing non-linear decision surfaces in the input space. All kernel functions must satisfy Mercer's condition that corresponds to the inner product of some feature space. The type of SVM constructed is a function of the selected kernel function and affects the computation time of implementing the SVM.

Many works in forecasting have demonstrated the favorable performance of the radial basis function [1,19,20]as Kernel function for SVM. Therefore, the Radial Basis Function (RBF),exp$\left\{-\gamma|x-x_i|^2\right\}$ is adopted in this work. The selection of the three parameters γ, ε and C of SVM model influence the accuracy of forecasting. However there is no standard method of selection of these parameters. Therefore Particle Swarm Optimization technique is used in the proposed model to optimize parameter selection.

3 Quantum Behaved Particle Swarm Optimization Technique in Selecting the Parameters of SVM

PSO has been a popular method for the problem of global optimization of functions in continuous space primarily because of its simple implementation and less computational demands. However, its convergence to the global optimum is not guaranteed. Therefore a large number of improvements upon the original PSO have been proposed since its development. A Quantum-behaved Particle Swarm Optimization (QPSO), inspired from quantum mechanics and based on delta-potential-well model, was proposed by Sun, Xu and Feng [17]. The salient features of QPSO, compared to PSO, include a reduction in the number of parameters and superior performance on a variety of functions in multi-dimensional continuous space.

QPSO was inspired by analysis of the convergence of the traditional PSO and quantum system. In the quantum physics, the state of a particle with momentum and energy can be depicted by its wave function $\psi(x,t)$. According to QPSO theory each particle is in a quantum state and is formulated by its wave function $\psi(x,t)$ instead of the position and velocity which are in PSO. According to the statistical significance of the wave function, the probability of a particle's appearing in a certain position can be obtained from the probability density function $|\ \psi(x,t)\ |^2$. And then the probability distribution function of the particle's position can be calculated through the probability density function . By employing the Monte Carlo method, the particle's position is updated according to the following equation:

$$X_{ij}^{t+1} = p_{ij}^{t} \pm 0.5 \cdot L_{ij}^{t} \cdot \ln(1/u_{ij}^{t}) \tag{7}$$

where u_{ij}^{t} is a random number uniformly distributed in (0, 1); p_{ij}^{t} is the local attractor and defined as

$$p_{ij}^{t} = \varphi_{ij}^{t} \cdot P_{ij}^{t} + (1-\varphi_{ij}^{t}) \cdot P_{gj}^{t} \tag{8}$$

where φ_{ij}^{t} is a random number uniformly distributed in (0, 1). In parameter L_{ij}^{t} is evaluated by

$$L_{ij}^{t} = 2 \cdot \beta \cdot |\ p_{ij}^{t} - X_{ij}^{t}\ | \tag{9}$$

where parameter β is called the contraction-expansion (CE) coefficient, which can be tuned to control the convergence speed of the algorithms. Then we get the position update equation as

$$X_{ij}^{t+1} = p_{ij}^{t} \pm \beta \cdot |\ p_{ij}^{t} - X_{ij}^{t}\ | \cdot \ln(1/u_{ij}^{t}) \tag{10}$$

The PSO algorithm with position update equation 10 is called as quantum delta-potential-well-based PSO (QDPSO) algorithm. Keeping in view the vital position of L for convergence rate and performance of the algorithm an improvement was proposed to evaluate parameter L. As per this algorithm the mean best position (mbest) is defined as the center of pbest positions of the swarm. That is

$$\begin{aligned} mbest^{t} &= (mbest_1^{t}, mbest_2^{t}, \ldots, mbest_D^{t}) \\ &= \left(\frac{1}{M}\sum_{i=1}^{M} P_{i1}^{t}, \frac{1}{M}\sum_{i=1}^{M} P_{i2}^{t}, \frac{1}{M}\sum_{i=1}^{M} P_{i3}^{t}, \ldots, \frac{1}{M}\sum_{i=1}^{M} P_{ij}^{t}, \ldots, \frac{1}{M}\sum_{i=1}^{M} P_{iD}^{t} \right) \end{aligned} \tag{11}$$

where M is the population size and P_i is the personal best position of particle i. Further Parameter L is given by

$$L_{ij}^t = 2 \cdot \beta \cdot \mid mbest_j^t - X_{ij}^t \mid \tag{12}$$

Hence, the particle's position is updated according to the following equation:

$$X_{ij}^{t+1} = p_{ij}^t \pm \beta \cdot \mid mbest_j^t - X_{ij}^t \mid \cdot \ln(1/u_{ij}^t) \tag{13}$$

The PSO algorithm with equation 13 is called as quantum-behaved particle swarm optimization (QPSO). Pseudo code for implementing the QPSO is given below:

```
Initialize the population size, the positions, and the dimensions of the particles
for $t = 1 \rightarrow$ Maximum Iteration T do
    Compute the mean best position mbest
    $\beta = (1.0 - 0.5) \ldots (T - t)/T + 0.5$
    for $i = 1 \rightarrow$population size M  do
        if $f(x_i) < f(p_i)$ then
            $P_i = X_i$
        end if
        $P_g = min(pP_i)$
        for $j = 1 \rightarrow$ dimension D do
            $\varphi = rand(0,1); u = rand(0,1);$
            $p_{ij} = \varphi \cdot P_{ij} + (1 - \varphi) \cdot P_{gj}$
            if $rand(0,1) > 0.5$ then
                $X_{ij} = p_{ij} + \beta$
            else
                $X_{ij} = p_{ij} - \beta \cdot (mbest - X_{ij}) \cdot log(1/u)$
            end if
        end for
    end for
end for
```

The objective of QPSO is to identify optimal values of (C, ε, γ) so that the error between the predicted and target values is minimized. Thus the objective function becomes;

$$\text{minimize} \quad N = \sum_{i=1}^{i=k} \frac{1}{k}(RMSE_i) \tag{14}$$

$$\text{subject to} \begin{cases} RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} [(h_m)_i - (h_s)_i]^2} \\ \hfill C_{min} \le C \le C_{max} \\ \hfill \varepsilon_{min} \le \varepsilon \le \varepsilon_{max} \\ \hfill \gamma_{min} \le \gamma \le \gamma_{max} \end{cases}$$

where h_s is the simulated groundwater level value obtained from the SVM model. Thus h_s depends upon the appropriate selection of the SVM parameters. In this equation, the constraints denote search ranges for the parameters C, ε and γ.

4 Applications and Discussions

4.1 Description of the Study Area

The study region chosen is Rentachintala, a mandal or tehsil situated in Guntur district. The monthly average rainfall and groundwater level data of the study region is available from 1985 to 2004. The groundwater level data was obtained from the Central Groundwater Board where as the average rainfall data is available from Indian Meteorological Department, Pune. The average annual normal rainfall in this region is around 681 *m* and the major contribution of rainfall is from the south west monsoon during the months of July-October. About 59 % of the annual precipitation falls during the south west monsoon and 26 % during the north east monsoon.

Since 2001, there has been a failure of rainfall over successive years which has resulted into a decrease in the normal rainfall by 14 %. Thus, because of lack of surface water, the demand for groundwater has increased substantially, which has had a very serious impact in the depletion of the groundwater resources. Figure 1 depicts the variation of average groundwater table in the study area for the period 1985-2004. The data from November 1984 to December 2001 forms training and validation data as shown in figure 1 and the remaining data forms the testing data. It can be seen from the figure 1 that the groundwater table has decreased gradually due to an increasing demand of groundwater coupled with a decreasing trend in rainfall. Also, in the Rentachintala region there is an availability of 77 ha.m of water in the non command areas whereas the demand for ground water utilization is about 100 ha.m, hence the groundwater resources is overly exploited in the non command areas of Rentachintala Mandal.

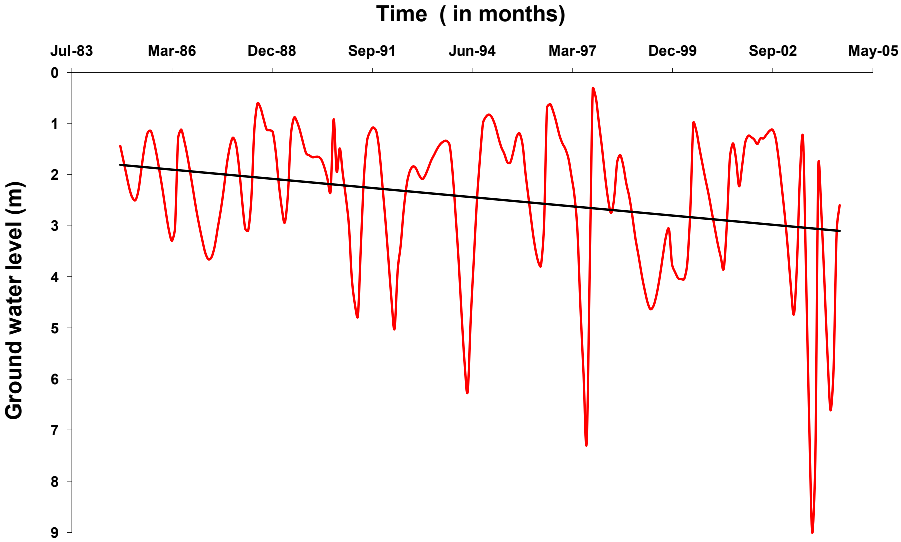


Fig. 1. Observed groundwater levels for the study period November 1984-October 2004

4.2 Performance Measures

Traditionally, the forecast performance is evaluated using the root mean square error (RMSE), Nash-Sutcliffe efficiency coefficient (EFF) and correlation coefficient (CORR). The evaluation criteria is determined by using the equation (Jain and Srinivasulu 2004):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}[(h_m)_i - (h_s)_i]^2} \tag{15}$$

$$EFF = 1 - \frac{\sum_{i=1}^{n}[(h_m)_i - (h_s)_i]^2}{\sum_{i=1}^{n}[(h_m)_i - (\bar{h}_m)]^2} \tag{16}$$

$$CORR = \sum_{i=1}^{n} \frac{[(h_m)_i - (\bar{h}_m)][(h_s)_i - (\bar{h}_s)]}{\sqrt{\sum_{i=1}^{n}[(h_m)_i - (\bar{h}_m)]^2 \sqrt{\sum_{i=1}^{n}[(h_s)_i - (\bar{h}_s)]^2}}} \tag{17}$$

Where h is the groundwater level and the subscripts $'m'$ and $'s'$ represent the measured and simulated values respectively. The average value of associated variable is represented with a 'bar' above it and n depicts the total number of training records.

4.3 Model Development

In this study, an appropriate input data set is identified by carefully analyzing the various combinations of the groundwater levels h at various time lags. The input vector is modified each time by successively adding an groundwater level at one more time lag leading to the development of a new SVM model.

As per the Central Ground Water Board (CGWB) reports the Groundwater in the region of Rentachintala is under confined conditions. The groundwater formations in this region lack primary porosity. However, it has developed secondary porosity through development of fractures and subsequent weathering over ages and has thus become water bearing. There is a possibility chance of rainwater percolating through these fractures and reaching the aquifer. This has been experimented in this study by including rainfall data ($R(t)$)to investigate the influence of Precipitation on groundwater in this region. Further the appropriate input vector is identified by comparing the coefficient of correlation, efficiency and root mean square error obtained by performing the 5 fold cross validation test for all the models given below. Five SVM models were developed with different set of inputs variables as follows .

Model 1 $h(t) = f[h(t-1)]$
Model 2 $h(t) = f[h(t-1), h(t-2)]$
Model 3 $h(t) = f[h(t-1), h(t-2), h(t-3)]$
Model 4 $h(t) = f[h(t-1), h(t-2), h(t-3), R(t-1)]$
Model 5 $h(t) = f[h(t-1), h(t-2), h(t-3), R(t-1), R(t-2)]$

Table 1 depicts the SVM parameters for the different models using Particle Swarm optimization technique (PSO) as mentioned in section 3 . Due to lack of any priori knowledge on the bounds of SVM parameters, a two-step PSO search algorithm [21] is recommended. In the first step, a coarse range search is made to achieve the best region of the three-dimensional grids. Since doing a complete grid-search may still be time-consuming, a coarse grid search is recommended first. After identifying a better region on the grid, a finer grid search on that region can be conducted. In the present study, the coarse range partitions for C are taken as $[10^{-5}, 10^{5}]$. Similarly, the coarse range partitions for ε are taken to be $[0, 10^{1}]$ and the coarse range partitions for γ are $[0, 10^{1}]$. Once the better region of grid is determined then a search is conducted in the finer range. Thus in the second stage search the parameter C ranges between $[10^{-1}, 10^{1}]$, ε is taken to be $[10^{-7}, 10^{-1}]$ and γ is taken to $[0, 1]$. The program terminates when the rmse values of the SVM model is less than 10^{-2} or it terminates after 1000 iterations. For real field cases it is difficult to obtained such low RMSE values so in general program terminates after 1000 iterations.

Table 1. Optimal SVM parameters obtained from PSO for different models

Model	C	ε	γ
Model 1	1.78	0.101	0.975
Model 2	1.056	0.017	0.6
Model 3	2.117	0.076	0.471
Model 4	1.692	0.111	0.935
Model 5	1.735	0.027	0.821

As seen from the table 2, the SVM models are evaluated based on their performance in the training and testing sets.The maximum coefficient of determination (r^2) obtained was 0.87 (in model 2) and the lowest r^2 term was 0.29 (model 4). In addition to this, model 2 exhibits the maximum value of efficiency (0.85) and minimum RMSE value (0.42). Model 2 which consists of two lags of groundwater level shows the highest efficiency, correlation and minimum RMSE. As a result, model 2 has been selected as the best-fit model to estimate the groundwater level in the Rentachintala region of the Guntur District.

Table 2. Optimal SVM parameters obtained from PSO for different models

Model	r^2	RMSE	Efficiency
Model 1	0.84	0.72	0.70
Model 2	0.87	0.42	0.85
Model 3	0.65	1.28	-0.32
Model 4	0.29	5.9	1.5789
Model 5	0.45	8.98	- 10.98

5 Comparison with Other Forecasting Models

The forecasting accuracy of the proposed SVM-QPSO model is compared with Neural networks. To set a direct comparison, the ANN model is trained using the same training data set ($h(t) = f[h(t-1), h(t-2)]$) as used for the SVM-QPSO. ANN model developed in this study has architecture of 2-13-1. The architecture of ANN is obtained by a trail and error procedure. Table 3 is showing the performance of comparison of SVM model with ANN. For the testing data, SVM-QPSO model has RMSE of 0.43 whereas it is 0.782 for ANFIS, 0.64 for ANN, and 1.936 for ARMA process. Thus SVM-PSO shows an improved performance compared to other models. Similarly the correlation coefficient also improves when one moves from ARMA, ANN, ANFIS to SVM-PSO. As SVM-PSO predicts the groundwater level values with a R value of 0.946 while ANFIS model exhibits R value of 0.793, ANN has R value of 0.85 and for ARMA model it is 0.769.

Table 3. Comparison of SVM-PSO performance with the other forecasting models for testing data set.

Performance measures	SVM-QPSO	ANN
RMSE	0.43	0.64
r^2	0.94	0.85
Effciency	0.84	0.81

The accuracy of the lowest water level prediction in a one month time is computed as the percentage Error of the deepest water level fluctuation in the validation period ($\%EDLF$) and is given by the equation

$$\%EDLF = \left[\frac{h_o - h_c}{h_o}\right] \times 100 \tag{18}$$

Where h_o is the observed deepest water level in the data set and h_c is the calculated water level corresponding to the observed deepest water level. Although all the models underestimate the lower groundwater levels, the SVM underestimate it by 7.5 % whereas 15 % by ANN. Observed deepest groundwater level is 9.0055 m and SVM predicts it as 9.0 m. Where as ANN predicts as 8.28 m. This shows that SVM-PSO model is able to predict the deepest groundwater level with more accuracy compared to ANN.

6 Summary and Conclusions

The accurate prediction of groundwater levels is extremely helpful of planning and management of water resources. Therefore, in this study an attempt is made to develop an efficient forecasting model for predicting groundwater levels in the Rentachintala region of Andhrapradesh, India. The accuracy of SVM-QPSO model has been investigated in the present study. The SVM-QPSO model is obtained by combining two methods QPSO and SVM. SVM conducts structural minimization rather than the minimization of the errors. Further QPSO selects the optimal SVM parameters to improve the forecasting accuracy. So this unique combination of SVM and QPSO has made the proposed SVM-QPSO model to performs better compared to the other models. Also a number of performance measures like coefficient of correlation, root mean square error (RMSE) and Nash sutcliffe efficiency were considered for comparing the performance of SVM-PSO with the other forecasting techniques. In all the cases the results indicate that SVM-QPSO model forecasts are more accurate in predicting the groundwater levels. Thus SVM-QPSO model can be better alternative for forecasting groundwater level at a particular region. Further, for future work the information like use of water for agricultural and other purposes can be collected and can obtain the reasons for dips in groundwater levels at certain period of years and those can form as good input parameters.

References

1. Sivapragasam, C., Liong, S., Pasha, M.: Rainfall and runoff forecasting with SSA-SVM approach. Journal of Hydroinformatics 3, 141–152 (2001)
2. Khadam, I., Kaluarachchi, J.: Use of soft information to describe the relative uncertainty of calibration data in hydrologic models. Water Resources Research 40 (2004) W11505
3. Asefa, T., Kemblowski, M., McKee, M., Khalil, A.: Multi-time scale stream flow predictions: The support vector machines approach. Journal of Hydrology 318, 7–16 (2006)
4. Yu, X., Liong, S., Babovic, V.: Hydrologic Forecasting with Support Vector Machine Combined with Chaos-Inspired Approach. In: Hydroinformatics 2002: Proceedings of the 5th International Conference on Hydroinformatics, pp. 1–5 (2002)
5. Wang, W., Chau, K., Cheng, C., Qiu, L.: A comparison of performance of several artificial intelligence methods for forecasting monthly discharge time series. Journal of Hydrology 374, 294–306 (2009)
6. Wu, C., Chau, K., Li, Y.: River stage prediction based on a distributed support vector regression. Journal of Hydrology 358, 96–111 (2008)
7. Wu, C., Chau, K., Li, Y.: Predicting monthly streamflow using data-driven models coupled with data-preprocessing techniques. Water Resources Research 45 (2009) W08432
8. Chau, K., Wu, C.: A hybrid model coupled with singular spectrum analysis for daily rainfall prediction. Journal of Hydroinformatics 12, 458–473 (2010)
9. Lin, J., Cheng, C., Chau, K.: Using support vector machines for long-term discharge prediction. Hydrological Sciences Journal 51, 599–612 (2006)
10. Cherkassky, V., Ma, Y.: Practical selection of SVM parameters and noise estimation for SVM regression. Neural Networks 17, 113–126 (2004)
11. Müller, K., Smola, A., Rätsch, G., Schölkopf, B., Kohlmorgen, J., Vapnik, V.: Using support vector machines for time series prediction (2000)
12. Schölkopf, B., Bartlett, P., Smola, A., Williamson, R.: Support vector regression with automatic accuracy control. In: Proceedings of ICANN 1998, Perspectives in Neural Computing, Citeseer, pp. 111–116 (1998)
13. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, 1995, vol. 4, pp. 1942–1948. IEEE (1995)

14. Hassan, R., Cohanim, B., De Weck, O., Venter, G.: A comparison of particle swarm optimization and the genetic algorithm. In: Proceedings of the 1st AIAA Multidisciplinary Design Optimization Specialist Conference (2005)
15. Clerc, M.: The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. In: Proceedings of the 1999 Congress on Evolutionary Computation, CEC 1999, vol. 3. IEEE (1999)
16. Zheng, Y., Ma, L., Zhang, L., Qian, J.: Empirical study of particle swarm optimizer with an increasing inertia weight. In: The 2003 Congress on Evolutionary Computation, CEC 2003, vol. 1, pp. 221–226. IEEE (2003)
17. Sun, J., Xu, W., Feng, B.: A global search strategy of quantum-behaved particle swarm optimization. In: 2004 IEEE Conference on Cybernetics and Intelligent Systems, pp. 111–116. IEEE (2004)
18. Cortes, C., Vapnik, V.: Support-vector networks. Machine learning 20, 273–297 (1995)
19. Choy, K., Chan, C.: Modelling of river discharges and rainfall using radial basis function networks based on support vector regression. International Journal of Systems Science 34, 763–773 (2003)
20. Yu, X., Liong, S., Babovic, V.: EC-SVM approach for real-time hydrologic forecasting. Journal of Hydroinformatics 6, 209–223 (2004)
21. Hsu, C., Chang, C., Lin, C., et al.: A practical guide to support vector classification (2003)

Genetic Algorithm Based Optimal Design of Hydraulic Structures with Uncertainty Characterization

Raj Mohan Singh

Department of Civil Engineering, Motilal Nehru National Institute of Technology,
Allahabad-211004, India
`rajm@mnnit.ac.in, rajm.mnnit@gmail.com`

Abstract. Hydraulic structures such as weirs or barrages are integral parts of diversion head works in the alluvial plains of India. However, there is no fixed procedure to design the basic barrage parameters. The depth of sheet piles, the length and thickness of floor may be treated as basic barrage parameters. Present work discusses the procedure of optimal design using Genetic algorithm (GA). An optimization based methodology is presented to obtain the optimum structural dimensions that minimize the total cost as well as satisfy the exit gradient criteria. Nonlinear optimization formulation (NLOF) with subsurface flow embedded as constraint in the optimization formulation is solved by GA. The results obtained in this study shows that considerable cost savings can be achieved when the proposed NLOF is solved using GA. Further, uncertainty in design, and hence cost from uncertain hydrogeologic parameter, seepage head, is investigated using fuzzy numbers.

Keywords: nonlinear optimization formulation, genetic algorithm, hydraulic structures, barrage design, fuzzy numbers, uncertainty characterization.

1 Introduction

A safe and optimal design of hydraulic structures is always being a challenge to water resource researchers. The seepage head causing the seepage vary with variation in flows. Design of hydraulic structures should also insure safety against seepage induced failure of the hydraulic structures. The variation in seepage head affects the downstream sheet pile depth, overall length of impervious floor, and thickness of impervious floor. The exit gradient, which is considered the most appropriate criterion to ensure safety against seepage induced piping [1, 2, 3, 4] on permeable foundations, exhibits non linear variation in floor length with variation in depth of down stream sheet pile. These facts complicate the problem and increase the non linearity of the problem. However, an optimization problem may be formulated to obtain the optimum structural dimensions that minimize the cost as well as satisfy the safe exit gradient criteria.

The optimization problem for determining an optimal section for the weirs or barrages consists of minimizing the construction cost, earth work, cost of sheet piling,

B.K. Panigrahi et al. (Eds.): SEMCCO 2011, Part I, LNCS 7076, pp. 742–749, 2011.
© Springer-Verlag Berlin Heidelberg 2011

and length of impervious floor [5, 6]. Earlier work [5] discussed the optimal design of barrage profile for single deterministic value of seepage head. This study first solve the of nonlinear optimization formulation problem (NLOP) using genetic algorithm (GA) which gives optimal dimensions of the barrage profile that minimizes unit cost of concrete work, and earthwork and searches the barrage dimension satisfying the exit gradient criteria. The work is then extended to characterize uncertainty in design due to uncertainty in measured value of exit gradient, an important hydrogeologic parameter. Uncertainty in design, and hence cost from uncertain safe exit gradient value are quantified using fuzzy numbers.

2 Subsurface Flow

The general seepage equation under a barrage profile may be written as:

$$\frac{\partial^2 h}{\partial x^2}+\frac{\partial^2 h}{\partial y^2}+\frac{\partial^2 h}{\partial z^2}=0 \tag{1}$$

This is well known Laplace equation for seepage of water through porous media. This equation implicitly assumes that (i) the soil is homogeneous and isotropic; (ii) the voids are completely filled with water; (iii) no consolidation or expansion of soil takes place;and (iv) flow is steady and obeys Darcy’s law.

For 2-dimensional flow, the seepage equation (1) may be written as:

$$\frac{\partial^2 h}{\partial x^2}+\frac{\partial^2 h}{\partial y^2}=0 \tag{2}$$

The need to provide adequate resistance to seepage flow represented by equation (1) both under and around a hydraulic structure may be an important determinant of its geometry [7]. The boundary between hydraulic structural surface and foundation soil represents a potential plane of failure.

Stability under a given hydraulic head could in theory be achieved by an almost limitless combination of vertical and horizontal contact surfaces below the structure provided that the total length of the resultant seepage path were adequately long for that head [7, 8]. Present work utilized Khosla's Method of independent variables [4] to simulate the subsurface behavior in the optimization formulation.

Method of independent variables is based on Schwarz-Christoffel transformation to solve the Laplace equation (1) which represents seepage through the subsurface media under a hydraulic structure. A composite structure is split up into a number of simple standard forms each of which has a known solution. The uplift pressures at key points corresponding to each elementary form are calculated on the assumption that each form exists independently. Finally, corrections are to be applied for thickness of floor, and interference effects of piles on each others.

An explicit check is for the stability of the hydraulic structure for soil at the exit is devised by Khosla [4]. The exit gradient for the simple profile as in Fig. 1 is given by as follows:

$$G_E = \frac{H}{d_d}\frac{1}{\pi\sqrt{\lambda}} \tag{3}$$

where $\lambda = \frac{1}{2}[1+\sqrt{1+\alpha^2}]$; $\alpha = \frac{L}{d_d}$; L is total length of the floor; and H is the seepage head.

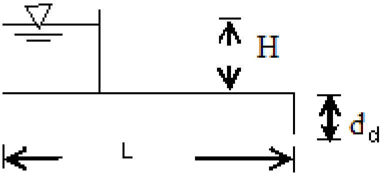


Fig. 1. Schematic of parameters used in exit gradient

Equation (3) gives G_E equal to infinity for no sheet pile at the downstream side of the floor. Therefore, it is necessary that a vertical cutoff (sheet pile) be provided at the downstream end of the floor. To prevent piping, the exit gradient is kept well below the critical values which depend upon the type of soil. The present work uses GA based optimization formulation incorporating uplift pressure and exit gradient in the optimization model to fix depth of sheet piles and length and thickness of floor. The optimization solution thus ensures safe structure with economy.

3 Optimal Design Formulation

3.1 Optimization Model

Minimize $C(L, d_1, d_d) = c_1(f_1) + c_2(f_2) + c_3(f_3) + c_4(f_4) + c_5(f_5)$ (4)

Subject to $$SEG \geq \frac{H}{d_d \pi\sqrt{\lambda}} \tag{5}$$

$$L^l \leq L \leq L^u \tag{6}$$

$$d_1^l \leq d_1 \leq d_1^u \tag{7}$$

$$d_d^l \leq d_d \leq d_d^u \tag{8}$$

$$L, d_1, d_d \geq 0 \tag{9}$$

where C (L, d_1, d_d) is objective function represents total cost of barrage per unit width (₹/m), ₹ is symbol for currency of INR (Indian National Rupee). The representation C is function of floor length (L), upstream sheet pile depth (d_1) and downstream sheet pile depth (d_d); f_1 is total volume of concrete in the floor per unit width for a given barrage profile and c_1 is cost of concrete floor (₹/m^3); f_2 is the depth of upstream sheet pile below the concrete floor and c_2 is the cost of upstream sheet pile including driving (₹/m^2); f_3 is the depth of downstream sheet pile below the concrete floor and c_3 is the cost of downstream sheet pile including driving (₹ /m^2); f_4 is the volume of soil excavated per unit width for laying concrete floor and c_4 is cost of excavation including dewatering (₹ /m^3); f_5 is the volume of soil required in filling per unit width and c_5 is cost of earth filling (₹ /m^3); SEG is safe exit gradient for a given soil formation on which the hydraulic structure is constructed and is function of downstream depth and the length of the floor; $\lambda = \frac{1}{2}[1+\sqrt{1+\alpha^2}]$; $\alpha = \frac{L}{d_d}$; L is total length of the floor; H is the seepage head ; d_1 is the upstream sheet pile depth; d_2 is downstream sheet pile depth; L^l, d_1^1, and d_d^1 is lower bound on L, d_1 and d_d respectively; L^u, $d_1{}^u$, $d_d{}^u$ are upper bound on L, d_1 and d_d respectively. The constraint equation (5) may be written as follows after substituting the value of λ :

$$L-d_d\left(\{2(\frac{H}{d_d\pi(SGE)})^2-1\}^2-1\right)^{1/2} \geq 0 \qquad (10)$$

In the optimization formulation, for a give barrage profile and seepage head H, f_1 is computed by estimating thickness at different key locations of the floor using Khosla's method of independent variables and hence nonlinear function of length of floor (L), upstream sheet pile depth (d_1) and downstream sheet pile depth (d_2). Similarly f_4, and f_5 is nonlinear. The constraint represented by equation (10) is also nonlinear function of length of the floor and downstream sheet pile depth (d_2). Thus both objective function and constraint are nonlinear; make the problem in the category of nonlinear optimization program (NLOP) formulation, which are inherently complex.

3.2 Characterizing Model Functional Parameters

For a given geometry of a barrage and seepage head H, the optimization model functional parameters f_1, f_2, f_3, f_4 and f_5 are characterized for the barrage profile shown in Fig. 2. Intermediate sheet-piles are not effective in reducing the uplift pressures and only add to the cost of in reducing the uplift pressures and only add to the cost of the barrage [5]. In present work, no intermediate sheet piles are considered.

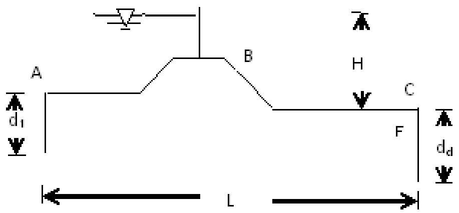


Fig. 2. Schematic of barrage parameters utilized in performance evaluation

The optimization model represented by equations (4)-(9) and the functional parameters embedded in the optimization model are solved using Genetic Algorithm on MATLAB platform. The basic steps employed in solution are available in Singh [6]. Table 1 shows physical parameters obtained by conventional methods for Fig. 2.

Table 1. Barrage profile parameters

Physical parameters	Values (meters)
*L	105.37
H	7.12
*d1	5.45
*d_d	5.9

* Decision variables to be optimized

3.3 Optimization Procedure Using Genetic Algorithm

GA was originally proposed by Holland [9] and further developed by Goldberg [10]. It is based on the principles of genetics and natural selection. GA's are applicable to a variety of optimization problems that are not well suited for standard optimization algorithms, including problems in which the objective function is discontinuous, non-differentiable, stochastic, or highly nonlinear [11]. The GA search starts from a population of many points, rather than starting from just one point. This parallelism means that the search will not become trapped on local optima [12].

The optimization model represented by equations (4)-(10) and the functional parameters embedded in the optimization model are solved using Genetic Algorithm on MATLAB platform. The basic steps employed in solution procedure may be presented as follows:

(i) Specification of parameters (decision variables) and hydrogeologic parameters (seepage head, and exit gradient) of problem domain in optimization formulation.
(ii) Randomly generate initial population of potential values of parameters in forms of strings.
(iii) Simulate seepage flow with decoded parameters to characterize f_1, f_2, f_3, f_4, and f_5 to evaluate objective function satisfying constraints.
(iv) Assign fitness value of each individual of population using objective function information.
(v) Stop if termination criteria satisfied, otherwise select and met the individual with high fitness value. More fit individual end up with more copies of themselves.
(vi) Perform cross-over operation on the selected parent population.
(vii) Perform mutation operation as in cross over operation with low probability.
(viii) Obtain new population after cross-over and mutation.
(ix) Go to step (iv).

In crossover, the offspring or children from the parents in the mating pool is determined. Mutation is performed with very low probability equal or close to the inverse of population size [13]. Such a low probability is helpful in keeping diversity in the population, and prevents the convergence of GA to local minima. The present work employed genetic algorithm code [14], and implemented it on MATLAB platform. The termination criteria is assumed to be satisfied when the population converges i.e. the average fitness of the population matches with the best fitness of the population and/or there is a little improvement in fitness with increase in number of generations.

4 Uncertainty Characterization in the Optimization Model

In the presence of limited, inaccurate or imprecise information, simulation with fuzzy numbers represents an alternative to statistical tool to handle parametric uncertainty. Fuzzy sets offer an alternate and simple way to address uncertainties even for limited exploration data sets. In the present work, the optimal design is first obtained assuming a deterministic value of hydrogelogic parameter, seepage head, in optimization model. Uncertainty in seepage head is then characterized using fuzzy numbers. The fuzzified NLOF is then solved using GA.

4.1 Uncertainty in Hydrogeologic Parameters

Uncertainty characterization of hydrogeologic parameters such as exit gradient and seepage head is based on Zadeh's extension principle [15]. In this study only seepage head is considered to be imprecise. Input exit gradient, as imprecise parameter, is represented by fuzzy numbers. The resulting output i.e. minimum cost obtained by the optimization model is also fuzzy numbers characterized by their membership functions. The reduced TM [16] is used in the present study. The measure of uncertainty used is the ratio of the 0.1-level support to the value of which the membership function is equal to 1 [17].

5 Results and Discussions

A typical barrage profile, a spillway portion of a barrage, is chosen for illustrating the proposed approach as shown in Fig. 2. The barrage profile shown in Fig. 2 and parameters values given in Table 1 is solved employing the methodology presented in this work. During the process of optimization, the process of going into new generation continues until the fitness of the population converged i.e. average fitness of population almost matches with the best fitness. The population converged at crossover rate of 0.5, mutation rate of 0.05, the population size is 40 and the number of generations is 100.

In the optimization approach the depth of sheet-piles determined from scour considerations is taken as a lower bound (3.0 m), and the upper bound is set from practical considerations and limited to 12.0 m. In present work, for performance evaluations, value of cost of concreting, c_1, is taken as ₹ 986.0/m^3; cost of sheet-piling including driving, c_1, is taken as ₹ 1510.0/m^2; cost of excavation and dewatering, c_3, is taken as ₹ 35.60/m^3; cost of earth filling, c_4, ia taken as ₹ 11.0/m^3. The optimized values of parameters for a deterministic seepage head value of 7.12m and safe exit gradient equal to 1/8 are shown in Table 2. It also resulted in a smaller floor length and overall lower cost. It has shown a savings in the barrage cost ranging from 16.73 percent. The GA based approach is also compared with classical optimization approach using non-linear constrained optimization function ‘FMINCON’ from MATLAB [18]. It can be seen from Table 3 that the reduction in cost is found to be more than six percent.

Table 2. Optimized parameters for safe exit gradient equal to 1/8

Method	Physical parameters Values			Cost ,₹ /m
	L	d1	d_d	
Conventional	105.37	5.45	5.9	133605.00
GA	61.00	3.1	9.2	111250.00

Table 3. Optimized parameters for safe exit gradient equal to 1/7

Optimization Method	Physical parameters			Cost, ₹ /m
	L	d1	d_d	
Classical optimization	51.61	5.45	10.42	111418.00
GA	40.36	9.16	9.81	104340.00

Earlier work discussed uncertainty characterization for fixed single range of value of uncertainty in seepage head. Earlier work [19] discussed uncertainty characterization for 15 percent uncertain seepage head. This work discusses three ranges of seepage head variations as 10 percent, 15 percent and 20 percent uncertainty in seepage head with central value of 7.12m. Each uncertain seepage head is represented in intervals by triangular fuzzy numbers. Safe exit gradient for this work is assumed to be deterministic with a fixed value of 0.175. The measure of uncertainty is estimated employing methods discussed in methodology section. The measure of uncertainty for three different uncertain seepage head scenario is found to be 16 percent, 22 percent and 31 percent respectively.

6 Conclusions

The present work also demonstrates the fuzzy based framework for uncertainty characterization in optimal cost for imprecise hydrologic parameter such as seepage head represented as interval fuzzy numbers. Left and right spread from central value of seepage head of 7.12 m is 10 percent, 15 percent and 20 percent and corresponding uncertainty in cost and hence design is 16 percent, 22 percent and 31 percent respectively. It may be concluded that uncertainty in seepage head reflects uncertain design with approximately 1.5 times more than the uncertainty in seepage head. The GA based optimization approach is equally valid for optimal design of other major hydraulic structures.

References

1. Khosla, A.N.: Pressure pipe observations at panjnad weir. Paper No. 160. Punjab Engineering Congress, Punjab, India (1932)
2. Khosla, A.N., Bose, N.K., Taylor, E.M.: Design of weirs on permeable foundations. CBIP Publication No. 12, Central Board of Irrigation and Power, New Delhi (1936) (reprint 1981)
3. Varshney, R.S., Gupta, S.C., Gupta, R.L.: Theory and design of irrigation structures, Nem Chand, Roorkee, vol. 2 (1988)
4. Asawa, G.L.: Irrigation and water resources engineering. New Age International, Limited Publishers, New Delhi (2005)
5. Garg, N.K., Bhagat, S.K., Asthana, B.N.: Optimal barrage design based on subsurface flow considerations. Journal of Irrigation and Drainage Engineering 128(4), 253–263 (2002)
6. Singh, R.M.: Optimal design of barrages using genetic algorithm. In: Proceedings of National Conference on Hydraulics & Water Resources (Hydro 2007), SVNIT, Surat, pp. 623–631 (2007)
7. Skutch, J.: Minor irrigation design DROP - Design manual hydraulic analysis and design of energy-dissipating structures. TDR Project R 5830, Report OD/TN 86 (1997)
8. Leliavsky, S.: Irrigation engineering: canals and barrages. Oxford and IBH, New Delhi (1979)
9. Holland, J.H.: Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor (1975)
10. Goldberg, D.E.: Genetic algorithms in search, optimization and machine learning. Kluwer Academic Publishers, Boston (1989)
11. Haestad, M., Walski, T.M., Chase, D.V., Savic, D.A., Grayman, W., Beckwith, S., Koelle, E.: Advanced water distribution modeling and management, pp. 673–677. Haestad Press, Waterbury (2003)
12. Singh, R.M.: Identification of unknown pollution sources in groundwater using artificial neural networks and genetic algorithm. Ph.D. thesis, IIT Kanpur (2004)
13. De Jong, K.A.: An analysis of the behavior of a class of genetic adaptive systems. Ph.D. dissertation, Univ. of Mich., Ann Arbor (1975)
14. Passino, K.M.: Biomimicry for optimization, control, and automation. Springer, London (2005)
15. Zadeh, L.A.: Fuzzy algorithms. Information and Control 12, 94–102 (1968)
16. Hanss, M., Willner, K.: On using fuzzy arithmetic to solve problems with uncertain model parameters. In: Proc. of the Euromech 405 Colloquium, Valenciennes, France, pp. 85–92 (1999)
17. Abebe, A.J., Guinot, V., Solomatine, D.P.: Fuzzy alpha-cut vs. Monte Carlo techniques in assessing uncertainty in model parameters. In: 4th Int. Conf. Hydroinformatics, Iowa, USA (2000)
18. Singh, R.M.: Design of barrages with genetic algorithm based embedded simulation n optimization approach. Water Resources Management 25, 409–429 (2011)
19. Singh, R.M.: Optimal hydraulic structures profiles under uncertain seepage head. In: World Renewable Energy Congress (WREC 2011), May 8-13. Linköping University, Sweden (2011)

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, Swagamtam 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 II-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, Dusmanta 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
Raghuwanshi, M.M. I-265
Raha, Souvik I-102
Rahmani, Adel T. II-201
Raj, M. Victor I-323
Rajan, C. Christober 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
Sirisetti, 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